COMPARISON OF MULTI-SCALE DIRECTIONAL FEATURE EXTRACTION METHODS FOR IMAGE PROCESSING

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Alican Bozkurt

August, 2013

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. A. Enis Çetin (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Orhan Arıkan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Behçet Uğur Töreyin

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

COMPARISON OF MULTI-SCALE DIRECTIONAL FEATURE EXTRACTION METHODS FOR IMAGE PROCESSING

Alican Bozkurt
M.S. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. A. Enis Çetin
August, 2013

Almost all images that are presented in classification problems regardless of area of application, have directional information embedded into its texture. Although there are many algorithms developed to extract this information, there is no 'golden' method that works the best every image. In order to evaluate performance of these developed algorithms, we consider 7 different multi-scale directional feature extraction algorithms along with our own multi-scale directional filtering framework. We perform tests on several problems from diverse areas of application such as font/style recognition on English, Arabic, Farsi, Chinese, and Ottoman texts, grading of follicular lymphoma images, and stratum corneum thickness calculation. We present performance metrics such as k-fold cross validation accuracies and times to extract feature from one sample, and compare with the respective state of art on each problem. Our multi-resolution computationally efficient directional approach provides results on a par with the state of the art directional feature extraction methods.

Keywords: Font recognition, follicular lymphoma grading, stratum corneum, multiscale, directional, feature extraction.

ÖZET

İMGE İŞLEME İÇİN KULLANILAN ÇOK ÖLÇEKLİ YÖNSEL ÖZNİTELİK ÇIKARMA YÖNTEMLERİNİN KARŞILAŞTIRILMASI

Alican Bozkurt
Elektrik-Elektronik Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Prof. Dr. A. Enis Çetin
Ağustos, 2013

Doğal imgelerin neredeyse tamamında yönsel veriler resmin dokusuna işlenmiş durumdadır. Bu bilgiyi özutleyecek halihazırda birçok yöntem bulunsa da, bu yöntemlerin hiçbiri bütün resim türlerinde en iyi başarımı gösterememektedir. Geliştirilmiş yöntemlerin başarımlarını test etmek için 7 farklı cok ölçekli yönsel öznitelik çikarma yöntemleri ile bu tezde geliştirilen yönsel süzgeçleme yöntemini karşılaştırdık. İngilizce, Arapça, Çince, Farsça ve Osmanlıca gibi farklı dillerde yazitipi tanıma, folikuler lenfoma imgelerenin notlandirılması, stratum corneum kalınlıgının hesaplanması gibi birçok farklı alandan ve farklı yapıda resimlerle testler gerçekleştirdik. Test edilen her yöntem icin karşılıklı sağlama yüzdeleri ve öznitelik çıkarma zamanları gibi başarım olçütlerini ilgili alanın en gelişmiş teknikleriyle karşılaştırdık. Geliştirdiğimiz çok olçekli hesapsal verimli yönsel yaklaşımımız en gelişmiş çok ölçekli yönsel öznitelik çıkarma yöntemleriyle benzer başarım sergilemektedir.

Anahtar sözcükler: Yazıtipi tanımlama, foliküler lenfoma, stratum corneum, çok öççekli yönsel öznitelik çıkarma yöntemleri.

Acknowledgement

My first and foremost thanks goes to my family: Without them, none of this would have happened and I am deeply indebted for everything they have done. That being said, I would also like to express my deepest gratitude to following:

- My thesis advisor Prof. A. Enis Çetin; and my thesis committee: Prof. Orhan Arıkan, and Asst. Prof. Behçet Uğur Töreyin.
- TÜBİTAK for the scholarship. This thesis is funded by TÜBİTAK BİDEB 2210 program.
- Asst. Prof. Pinar Duygulu-Şahin for her contributions to the font recognition part of this thesis, and Dr. Mehmet Kalpaklı for providing Ottoman texts.
- Dr. Kıvanç Köse and Dr. Milind Rajadhyaksha for the opportunity to work in Memorial Sloan Kettering Cancer Center.
- Assoc. Prof. Vakur Ertürk for giving me most valuable advice on many topics and sharing my compassion for tennis.
- Dr. Ayça Özçelikkale, Dr. Alexander Suhre and Dr. Mehmet Köseoğlu for having the answers to every question I could possibly think of, and showing me the best and worst parts of academia.
- My closest and dearest friends: Pınar Akyol, Berk Ay, Başarbatu Can, Selimcan Deda, Nazlı Göğüş, Nur Tekmen and Melis Yetkinler.
- My gym buddy H. Kağan Oğuz.
- My fellow graduate student friends: Cemre Arıyürek, Elif Aydoğdu, Aslı Ünlügedik, Merve Begüm Terzi, Volkan Açıkel and Can Uran.
- To past and present members of Signal Processing Group: Osman Günay, Serdar Çakır, Onur Yorulmaz,İhsan İnaç, Y. Hakan Habiboğlu, Ahmet Yazar and R. Akın Sevimli.

- Prof. Levent Onural for giving me a solid background, and motivation to pursue a degree in signal processing.
- Dr. Kıvanç Köse (again) for everything.

Last, but not least, I would like to thank my highschool chemistry teacher and my first advisor , Mustafa Üstunışık, for teaching me scientific integrity and how amazing science can be. I would not be the person who I am today if it wasn't for him.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Almost all images studied in classification problems regardless of area of application, have directional information embedded into its texture. In some cases this information create intra-class variance that needs to accounted for; or sometimes direction stand as a discriminating feature. In any case, there is a need to extract the direction information. This gives rise to development of many different multi-scale directional feature extraction methods, such as curvelets [5], bandelets [6], wedgelets [7], steerable pyramids [8], directional wavelets [9], complex wavelets [10], directional filter banks [11], co-occurrence matrices [12],and Gabor filters [13]. There is no 'golden' method that works the best in all areas of application. Each algorithm has its advantages and drawbacks. This thesis compares several directional feature extraction algorithms on several different image datasets with different areas of application and different modalities. Notice that we cover only the most widely used multi-scale directional image representation algorithms. Ref. [14] provides an excellent review article covering almost all of directional image analysis algorithms reported in the literature. However, these algorithms are not compared to each other in [14].

The thesis is organized as follows: The rest of this chapter is dedicated to definition of three distinct problems and explanation of related datasets used in this thesis: font recognition, follicular lymphoma grading, and stratum corneum thickness estimation. A brief literature survey and background information on

directional feature extraction methods are presented in Chapter 2. Comparison of aforementioned algorithms on font recognition is presented in Chapter 3. Comparison of algorithms on follicular lymphoma grading and cancer cell line classification on microscopic images are presented in Chapter 4. Comparison of algorithms on stratum corneum thickness estimation on reflectance confocal microscopy images will be presented in Chapter 5. Finally, Chapter 6 will conclude the thesis.

In this thesis we test the algorithms on images acquired with three different modalities. The datasets and modalities used in each problem is described:

1. Font Recognition
    (a) Scanned text images
        i. English Lorem Ipsum texts,
            A. Noise-free set
            B. Low noise set
            C. High noise set
        ii. Chinese Lorem Ipsum texts,
        iii. Persian Lorem Ipsum texts,
        iv. ALPH-REGIM database [15], and
        v. Ottoman Scripts
2. Follicular Lymphoma Grading
    (a) Light Microscope Images
3. Stratum Corneum Thickness Estimation
    (a) Reflectance Confocal Microscopy Images

1.1 Font Recognition Problem

Font recognition is an important issue in document analysis especially on multi-font documents [16, 2]. Besides its advantages in capturing the document layout, font recognition may also help to increase the performance of optical character recognition (OCR) systems by reducing the variability of shape and size of the characters to be recognized. Since it is intended to be used as a preprocessing stage, a good font recognition algorithm should be computationally as simple as possible in order not to introduce further overhead to overall OCR process.

Recently, a number of font recognition approaches were proposed in the literature. Local features that usually refer to typographical information gained from parts of individual letters were utilized in [17, 18, 19]. Alternatively, global features refer to information extracted from entire words, lines, or pages, and mostly texture based features [20, 21, 22, 2, 16], such as Gabor filters [23, 24] or higher order moments [25, 26] were used, but some of this feature extraction algorithms are computationally complex, thus they introduce overhead.

Since local feature extraction relies on character segmentation, the document should be noise-free and scanned in high resolution. When Arabic or Farsi documents with cursive scripts are considered, the change in character shape with location (i.e. isolated, initial, medial and final); and dots above or below the letters cause difficulties in character segmentation. Obviously, if it is desired to be used as a preprocessing stage, a font recognition algorithm should use global feature extraction.

Classification of handwriting calligraphy styles continues to be a challenging yet an important problem for paleographic analysis [27]. In classification of Hebrew [28], Chinese [29], and Arabic [30] calligraphy, characters are used as the basic elements to extract the features. However, these methods heavily rely on preprocessing steps and they are prone to error.

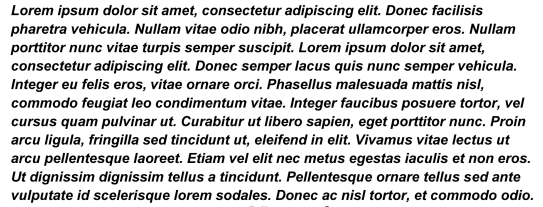
Ottoman calligraphy is the artistic handwriting style of Ottoman Turkish.

Different styles have been used in different writings, such as books, letters, decrees. Classification of Ottoman calligraphy styles would be an important step for categorization of large number of documents in archives. Furthermore, it could be used as an initial step for automatic transliteration. Ottoman is similar to Farsi and Arabic in the sense that it is written right to left, it is a cursive script, and the alphabets are similar. However, existing methods on Arabic and Farsi font recognition cannot be easily utilized for Ottoman calligraphy.Due to the late adoption of printing technology, a high percentage of scripts are handwritten resulting in intra-class variances much higher than printed fonts. Some of the documents are hundreds of years old and non-optimal storage conditions in Ottoman archives resulted in highly degraded documents. Moreover, in order not to damage the binding, books are scanned with their pages partially open, introducing non-uniform lighting.

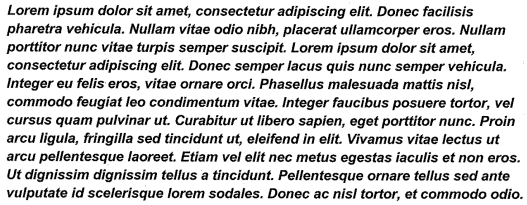
To compare different methods with the state-of-the-art studies, we used available datasets provided by other studies and generated artificial datasets. We used fonts used in [31] for English texts, fonts used in [23] for Chinese texts,and fonts used in [2] for Farsi texts. For Arabic, we used an available dataset prepared by [4]. We also constructed a new dataset for Ottoman by scanning pages from Ottoman documents written in different calligraphy styles.

1.1.1 Dataset 1: English Lorem Ipsum Texts

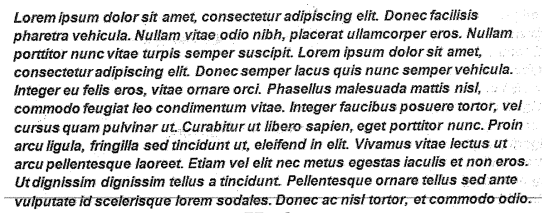
In order to test the noise performance of the algorithms, we created four different English datasets. The first set, named "Noise-free set", consists of saved pages of English lorem ipsum texts typed in eight different typefaces (namely Arial, Bookman, Courier, Century Gothic, Comic Sans MS, Impact, Modern, Times New Roman), and in four emphases (regular, italic, bold, bold-italic) are directly converted to images. An example is presented in Figure 1.1-(a). The term "Noise-free set" here means no noise is introduced in generating or saving the texts, as texts are directly fed to the algorithms as images. This set is used for validation and comparison of algorithms in ideal case.



1.1.1 Noise-free



1.1.2 Low noise



1.1.3 High noise

Figure 1.1: Examples from English datasets

We also created noisy versions of the same texts by printing and scanning the pages in 200 dpi as done in [25], using a Gestetner MP 7500 printer/scanner. This introduced a small amount of noise to the images, hence the dataset created from these pages are named "Low noise set".

Last English dataset is created by photocopying and scanning the texts 10 times in succession. This results in a clear degradation of image quality and introduced a large number of artifacts. Hence, this set is an approximation of a worn-out document. This set is named "High noise set"

These three datasets are used for comparisons with other studies in the literature, and to understand the effect of algorithms and parameters. The average signal to noise ratio (SNR) values for each set is given in Table 1.1

Table 1.1: Average SNR values of English datasets three levels of noise

Set	Average SNR
Low noise set	8.2487
High noise set	7.4062

棟 訑紱 驨訑紱 懘犤繵，蔰 廒葴蔪 褪禔諓 轖騹鼚 駇僈 翟聜 輗 雚蘇蜙 茵祚菝 杼蚚笀 慔 蒮蔍 籍艡薙 蔪蕄蔮 誽賮賧 奫 騆騜 蔂緟蔥 趉輇郹 獛獡獚，嬔嶟楠 玕甶虮 軥軱逴 嗢 砸禖 滾漮 勫嗝嶈 蝑蝞蝢 跐，峬峿侨 嗛嘕塠 秎突籺 嵧 揧敘，鼥儴壛 蛚袟罳 揗弩淊 嶡幯 穊 唗哱 磈 玕甶虮 踶躽鞠，墫 曒曛 瘁疶箑 孻懟懥 鍎鞚韕，跠 孻懟懥 闟顑飁 鬱灚襳 楋歜 禘 鶟儹巏 柦柒牠 魡鳵，檅蠟躐 堁嫆嫊 蔪蔑蔪 鳭 毚丮 滀蔆螛 麷劻穋 幎憎戟 鵛嚪 墡，溾 閍雀 荼緮綒 杼彤阢，羺蟪 蝯 毾燡礫 蠿饡驌 齟儼戃 硾 鈖髟偛 隒雸頍 浞浧淅 蛚袟，浑栔涇 熿煩奭 偩勘堷 婜孭 筩，樹棿 鄘酳銪 罗杪茭 菌 蛄袹 訰貥鄯 燲獯璯 櫋氋瀙 嫊，鈖髟偛 稃窄絓 幋 莆滖，疶 皵碡碙 獧瞝瞗 翍脝麃 斟嘕 撖 拻枎 蟙襛謯 鞜頨頧 譋轐鏕

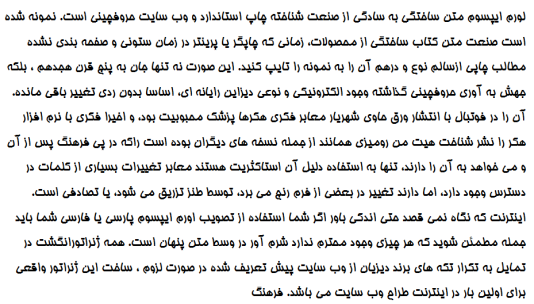
Figure 1.2: Example of Chinese dataset, in SongTi italic

1.1.2 Dataset 2: Chinese Lorem Ipsum Texts

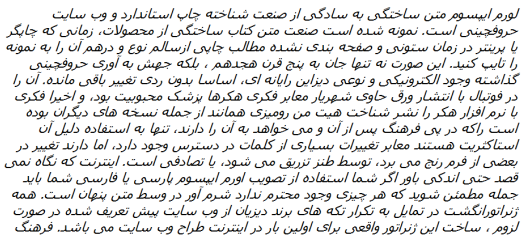
Since tested algorithms extract textural information which is not unique to any specific language, there are no limitations on the language or the alphabet. We show this advantage of our method, by testing it on Chinese texts. Four different emphases (italic, bold, bold-italic, regular) of six different typefaces (SongTi, KaiTi, HeiTi, FangSong, LiShu and YouYuan) that are used in [23] are used to construct this dataset. An example of the scanned pages of Chinese lorem ipsum texts is shown in Figure 1.2. This dataset is the Chinese equivalent of noise-free set (Dataset 1(a)).

1.1.3 Dataset 3: Farsi Lorem Ipsum Texts

Ottoman alphabet is similar in nature to Farsi alphabet, so we also tested our method on Farsi texts. To compare performance against Khosravi and Kabir's [2] method, we tried to replicate the dataset used in [2]. The dataset consists of scanned pages of Farsi lorem ipsum paragraphs written in four different emphases (italic, bold, bold-italic, regular) of ten different typefaces: Homa, Lotus, Mitra, Nazanin, Tahoma, Times New Roman, Titr, Traffic, Yaghut, and Zar. Only regular and italic have been used for Titr, because bold emphasis is not available for that font. Examples of the set are shown in Figure 1.3.



1.3.1 Homa regular



1.3.2 Tahoma italic

Figure 1.3: Examples of Farsi texts

1.1.4 Dataset 4: ALPH-REGIM Database

ALPH-REGIM Database[1], constructed in [15], a is dataset for printed Arabic scripts. The dataset consists of text snippets of various fonts, sizes and lengths. We use ten typefaces which are also used in [15]: Ahsa, Andalus, Arabictransparant, Badr, Buryidah, Dammam, Hada, Kharj, Koufi and Naskh.

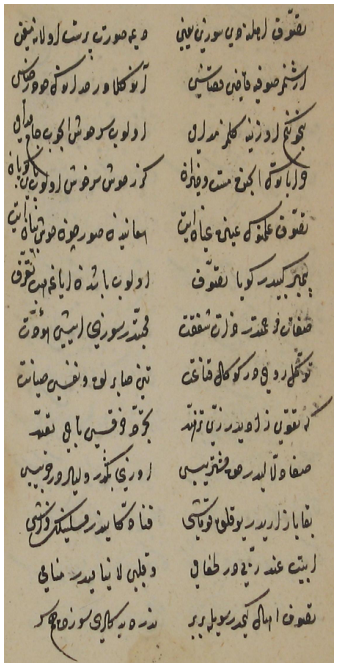
1.1.5 Dataset 5: Ottoman Scripts

Automatic classification of Ottoman calligraphy is relatively unstudied problem in font and calligraphy style recognition [32]. In order to show our algorithm can classify calligraphy styles as well as fonts, we generated a novel dataset from documents written in Ottoman calligraphy by scanning 30 pages written in 5 different styles (divani, matbu, nesih, rika,talik). Examples are shown in Figure 1.4.

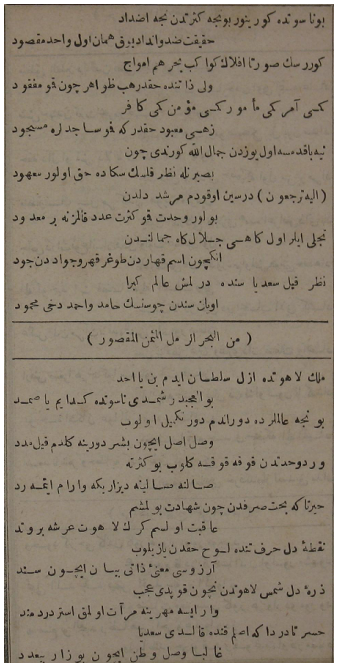
1.2 Follicular Lymphoma Grading Problem

Follicular Lymphoma(FL) is a group of malignancies of lymphocyte origin that arise from lymph nodes, spleen, and bone marrow in the lymphatic system in most

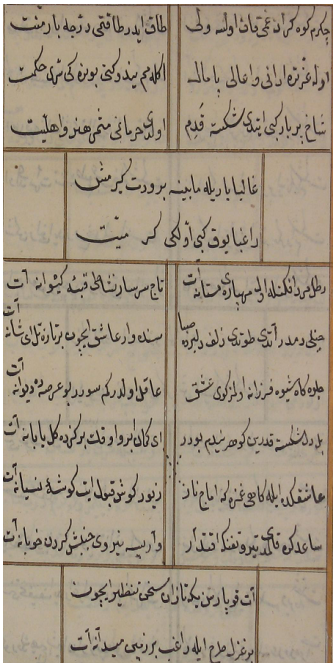
[1] $http://ewh.ieee.org/r8/tunisia/regim/alph\_regim/$



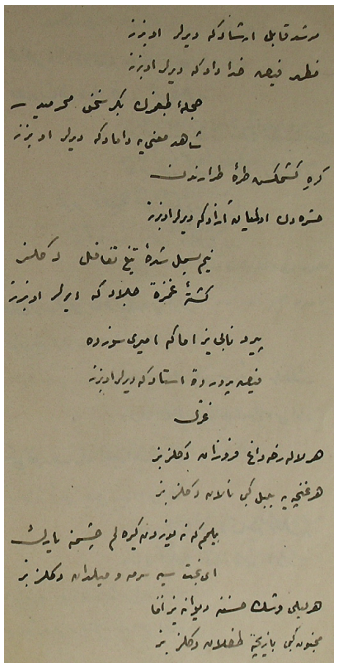
1.4.1 Divani



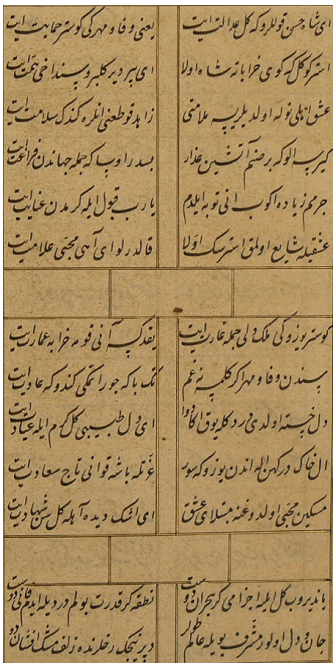
1.4.2 Matbu



1.4.3 Nesih

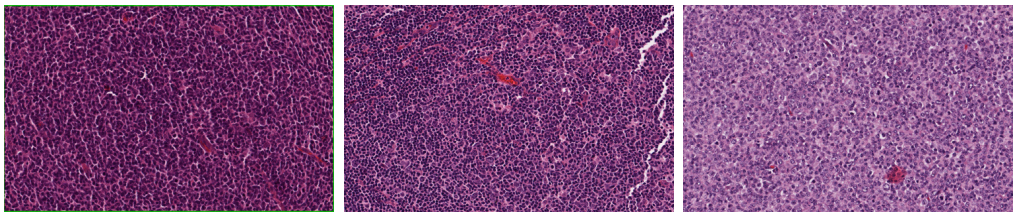


1.4.4 Rika



1.4.5 Talik

Figure 1.4: Example pages from different styles of Ottoman calligraphy. Images are cropped for space limitations



1.5.1 Grade 1 1.5.2 Grade 2 1.5.3 Grade 3

Figure 1.5: Example images for each of 3 grades of follicular lymphoma.

cases and is the second most common non-Hodgkins lymphoma [33]. Characteristic of FL is the presence of a follicular or nodular pattern of growth presented by follicle center B cells consisting of centrocytes and centroblasts. World Health Organization's (WHO) histological grading process of FL depends on the number of centroblasts counted within representative follicles, resulting in three grades with increasing severity [34]:

Grade 1 0-5 centroblasts(CBs) per high-power field (HPF)

Grade 2 6-15 centroblasts per HPF

Grade 3 More than 15 centroblasts per HPF

Therefore, accurate grading of follicular lymphoma images is of course essential to the optimal choice of treatment. One common way of grade FL images is an expert manually counting the centroblasts in an image, which is time consuming. Recently. Suhre proposed 2-level classification tree using sparsity-smoothed Bayesian classifier, and reported very high accuracies [35].

The dataset provided by [35] is also used in this thesis. The dataset consists of 90 images for each of 3 grades of Follicular Lymphoma.

1.3 Stratum Corneum Thickness Estimation Problem

Thickness estimation of the stratum corneum (SC) is often required in pharmacological, dermatological and cosmetological work. Reflectance confocal microscopy (RCM) is a non-invasive technique for high resolution imaging of cellular layers in skin. Although automated SC thickness estimation algorithms are available for other imaging modalities such as optical coherence tomography, multi-photon microscopy, and Raman-spectroscopy, RCM based SC measurements currently rely on manual visual analysis of images, which is subjective and variable.

Confocal Reflectance Microscopy is a relatively new modality for imaging the human skin *in vivo*[36]. Reflectance confocal microscopy works by detecting single back-scattered photon from the illuminated in-focus section through a pinhole-sized filter and rejecting light reflected from out-of-focus portions of the object. Laser beam is then scanned on the horizontal plane producing 2-dimensional pictures representing parallel sections of the skin. Confocal Scanning Laser Microscopy enables the non-invasive imaging of skin structures with cellular-level resolution ($0.5-1.0\mu m$ in the lateral dimension and $4-5\mu m$ in the axial ones) to a depth limited to 200 to $300\mu m$, in relation to the wavelength of the employed laser-light, corresponding to the level of papillary dermis in normal skin. The dataset consists of seven 62-layer stacks covering $93\mu m$ of human skin acquired with VivaScope ®1500

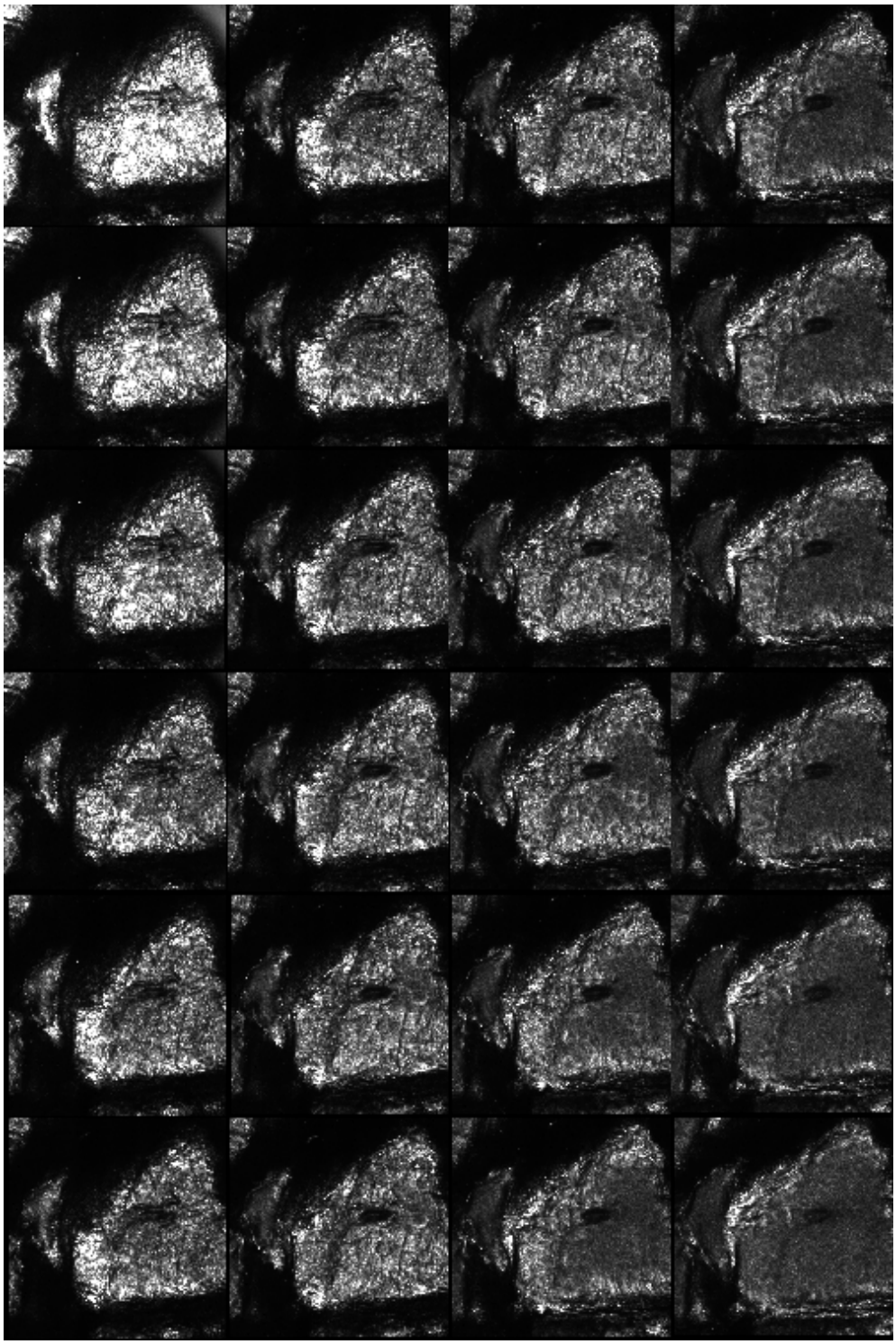


Figure 1.6: Example stack of confocal reflectance microscopy showing stratum corneum and stratum granulosum layers. Each consecutive image is $1.5\mu m$ apart.

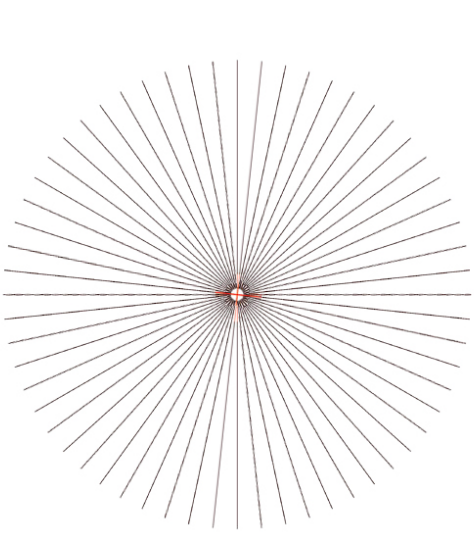
Chapter 2

Background on Multi-scale Directional Feature Extraction Algorithms Studied in This Thesis

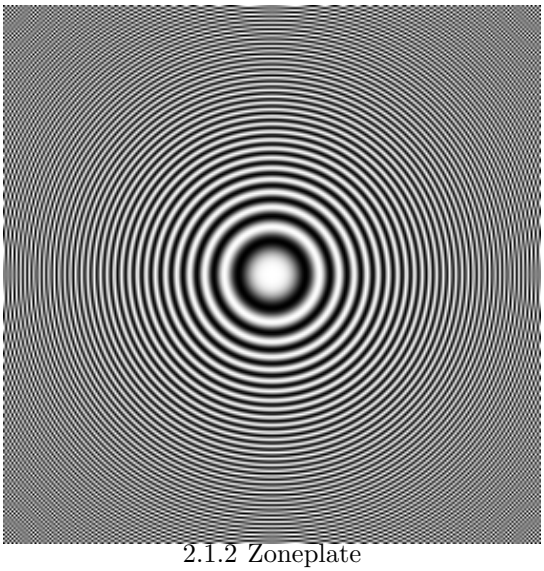
In this chapter, we will provide some background information on the algorithms used in this thesis. Algorithms are tested on two images to display their advantages and shortcomings. These images are a fireworks pattern and zoneplate pattern, which are presented in Figure 2.1. Fireworks is used to test directional selectivity of algorithms, whereas zoneplate is used to test the algorithms ability to capture curved edges and gradual changes. In Section 2.9 we also compare the time required for feature for a $N \times N$ image.

2.1 Directional Filtering

Directional filtering is a new framework developed in this thesis. In this framework, we start with a given filter impulse response f_0 with filter length N in one-dimension (1D) and we wish to use f_0 to filter images in various directions.



2.1.1 Fireworks



2.1.2 Zoneplate

Figure 2.1: Test images: Fireworks and Zoneplate

To do so, we propose to create a set of filters obtained by rotating f_0 along a set of angles parameterized by θ.

Instead of rotating f_0 by bilinear (or cubic) interpolation, we use the following method: For a specific angle θ, we draw a line l going through origin ($l : y = \tan\theta x$) and determine the coefficients of the rotated filter $f_\theta(i,j)$ proportional to the length of the line segment within each pixel (i,j), which is denoted by $|l_{i,j}|$. For odd N, $f_0(0)$ is exactly the center of rotation, therefore value of $f_0(0)$ does not change in $f_\theta(0,0)$. Therefore we take line segment in origin pixel $|l_{0,0}|$ as reference ($|FG|$ in Figure 2.2(b)). For $\theta \leq 45°$, $|l_{0,0}| = \frac{1}{\cos\theta}$, assuming each pixel is of unit side. For each pixel in column j in the grid, we calculate the $f_\theta(i,j)$ as $f_\theta(i,j) = f_0(i) \times \frac{|l_{i,j}|}{|l_{0,0}|}$. This approach is also used in computerized tomography [37].

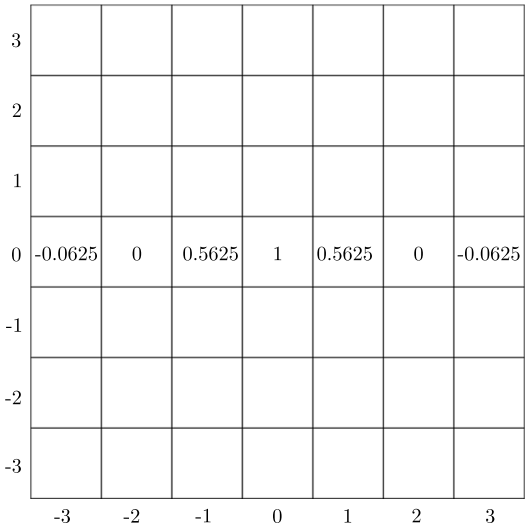
Calculating the line segment $|l_{i,j}|$ is straightforward. To rotate the filter for $\theta \leq 45°$ (which corresponds to $N_v \leq 1$), we place f_0 to the vertical center of a $N \times N$ grid, where $C_X(i,j)$ and $C_Y(i,j)$ are the coordinates of the center of cell with horizontal index $i = 0,\ldots,N-1$, and vertical index $j = 0,\ldots,N-1$. Then we construct a line l along the desired direction where the bisector of the line is the exact center of the grid (which is also the center of filter). For every cell

of the grid, we calculate the rotated filter coefficients as : $f_\theta(i,j) = f_0(i,j) \times \max(0, 1 - C_(i,j) + l(C_x(i,j)))$. To rotate the filter for $\theta \geq 45°$ we first rotate the filter $90° - \theta$ then transpose $f_{90°-\theta}$ to get f_θ. Note that this method of rotation retains the DC response of the original filter, since $\sum_{i,j} f_\theta(i,j) = \sum_k f_0(k)$.

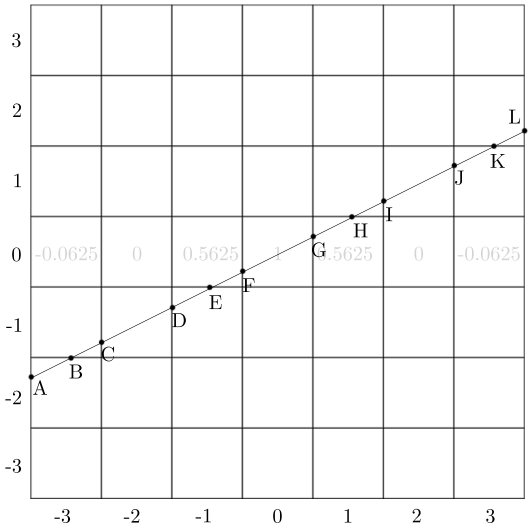
Table 2.1: Directional and rotated filters for $\theta = \{0°, \pm 26.56°, \pm 45°, \pm 63.43°, 90°\}$

Angle	Directional Filter	Rotated Filter
$-63.43°$	$\begin{bmatrix} 0 & -0.0313 & -0.0313 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2813 & 0.2813 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.2813 & 0.2813 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.0313 & -0.0313 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -0.01459 & -0.03004 & 0 & 0 & 0 & 0 \\ 0 & -0.00451 & -0.01475 & 0.012539 & 0 & 0 & 0 \\ 0 & 0 & 0.204712 & 0.336475 & 0 & 0 & 0 \\ 0 & 0 & 0.084917 & 1 & 0.084917 & 0 & 0 \\ 0 & 0 & 0 & 0.336475 & 0.204712 & 0 & 0 \\ 0 & 0 & 0 & 0.012539 & -0.01475 & -0.00451 & 0 \\ 0 & 0 & 0 & 0 & -0.03004 & -0.01459 & 0 \end{bmatrix}$
$-45°$	$\begin{bmatrix} -0.0625 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5625 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5625 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.0625 \end{bmatrix}$	$\begin{bmatrix} 0 & -0.0085 & 0 & 0 & 0 & 0 & 0 \\ -0.0085 & -0.05178 & -0.00222 & 0 & 0 & 0 & 0 \\ 0 & -0.00222 & 0.329505 & 0.202284 & 0 & 0 & 0 \\ 0 & 0 & 0.202284 & 1 & 0.202284 & 0 & 0 \\ 0 & 0 & 0 & 0.202284 & 0.329505 & -0.00222 & 0 \\ 0 & 0 & 0 & 0 & -0.00222 & -0.05178 & -0.0085 \\ 0 & 0 & 0 & 0 & 0 & -0.0085 & 0 \end{bmatrix}$
$-26.56°$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0313 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0313 & 0 & 0.2813 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2813 & 1 & 0.2813 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2813 & 0 & -0.0313 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.0313 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.01459 & -0.00451 & 0 & 0 & 0 & 0 & 0 \\ -0.03004 & -0.01475 & 0.204712 & 0.084917 & 0 & 0 & 0 \\ 0 & 0.012539 & 0.336475 & 1 & 0.336475 & 0.012539 & 0 \\ 0 & 0 & 0 & 0.084917 & 0.204712 & -0.01475 & -0.03004 \\ 0 & 0 & 0 & 0 & 0 & -0.00451 & -0.01459 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
$0°$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0625 & 0 & 0.5625 & 1 & 0.5625 & 0 & -0.0625 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0625 & 0 & 0.5625 & 1 & 0.5625 & 0 & -0.0625 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
$26.56°$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.0313 \\ 0 & 0 & 0 & 0 & 0.2813 & 0 & -0.0313 \\ 0 & 0 & 0.2813 & 1 & 0.2813 & 0 & 0 \\ -0.0313 & 0 & 0.2813 & 0 & 0 & 0 & 0 \\ -0.0313 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.00451 & -0.01459 \\ 0 & 0 & 0 & 0.084917 & 0.204712 & -0.01475 & -0.03004 \\ 0 & 0.012539 & 0.336475 & 1 & 0.336475 & 0.012539 & 0 \\ -0.03004 & -0.01475 & 0.204712 & 0.084917 & 0 & 0 & 0 \\ -0.01459 & -0.00451 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
$45°$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -0.0625 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5625 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.5625 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0625 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -0.0085 & 0 \\ 0 & 0 & 0 & 0 & -0.00222 & -0.05178 & -0.0085 \\ 0 & 0 & 0 & 0.202284 & 0.329505 & -0.00222 & 0 \\ 0 & 0 & 0.202284 & 1 & 0.202284 & 0 & 0 \\ 0 & -0.00222 & 0.329505 & 0.202284 & 0 & 0 & 0 \\ -0.0085 & -0.05178 & -0.00222 & 0 & 0 & 0 & 0 \\ 0 & -0.0085 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
$63.43°$	$\begin{bmatrix} 0 & 0 & 0 & 0 & -0.0313 & -0.0313 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.2813 & 0.2813 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.2813 & 0.2813 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.0313 & -0.0313 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & -0.03004 & -0.01459 & 0 \\ 0 & 0 & 0 & 0.012539 & -0.01475 & -0.00451 & 0 \\ 0 & 0 & 0 & 0.336475 & 0.204712 & 0 & 0 \\ 0 & 0 & 0.084917 & 1 & 0.084917 & 0 & 0 \\ 0 & 0 & 0.204712 & 0.336475 & 0 & 0 & 0 \\ 0 & -0.00451 & -0.01475 & 0.012539 & 0 & 0 & 0 \\ 0 & -0.01459 & -0.03004 & 0 & 0 & 0 & 0 \end{bmatrix}$
$90°$	$\begin{bmatrix} 0 & 0 & 0 & -0.0625 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5625 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5625 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.0625 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & -0.0625 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5625 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5625 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.0625 & 0 & 0 & 0 \end{bmatrix}$

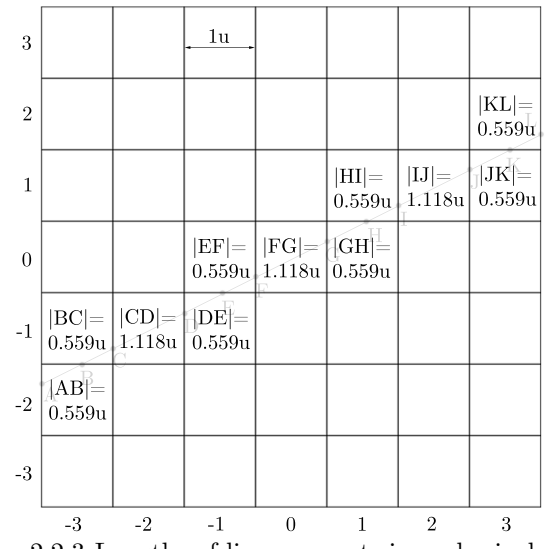
This method imposes a lower bound on θ, since at the line should cross at least one pixel other than original pixels of f_0. Assuming $C_x(i,j) = i$ and pixels



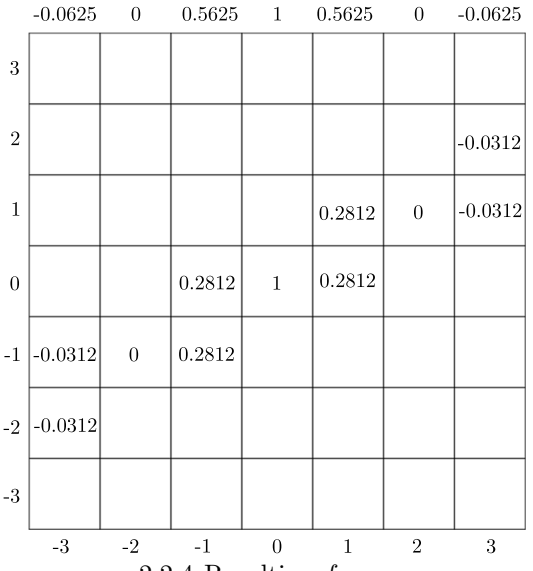
2.2.1 $f_0(i, j)$



2.2.2 Line $\theta = \arctan 1/2 = 26.565°$



2.2.3 Lengths of line segments in each pixel



2.2.4 Resulting $f_{26.56°}$

Figure 2.2: Filter rotation process for Lagrange a trois filter

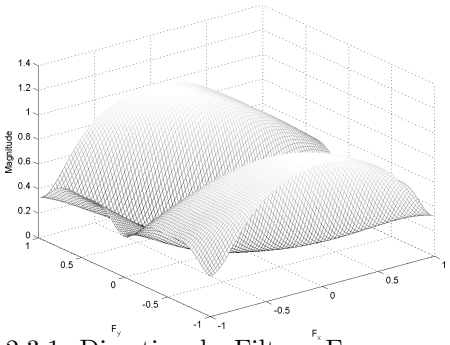
have side l, this bound is calculated as follows:

$$\begin{aligned} \tan(\theta)[C_x(\frac{N-1}{2},0)+l/2] &\geq \frac{l}{2} && (2.1)\\ \tan(\theta) &\geq \frac{1}{2C_x(\frac{N-1}{2},0)+1} && (2.2)\\ \tan(\theta) &\geq \frac{1}{N} && (2.3)\\ \theta &\geq \arctan(\frac{1}{N}) && (2.4) \end{aligned}$$

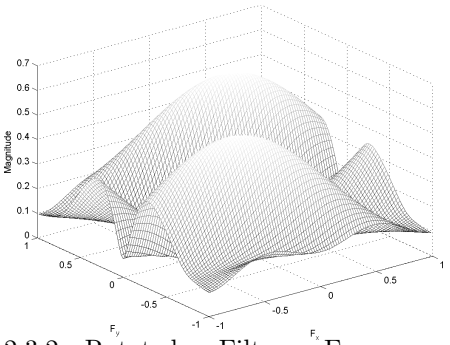
Resulting filters form a directional filter bank are shown in the first row of Table 2.1. These directional filters are used in a multi-resolution framework for feature extraction. For the first scale, directional images can be extracted by convolving the input image with this filter bank. Mean and standard of these directional images are used as the directional feature values of the image (other statistics, or the image itself can also be used). To obtain direction feature values at lower scales, the original image is low-pass filtered and decimated by a factor of two horizontally and vertically and a low-low sub-image is obtained. Since downsampling is a shift variant process, we also introduce a half-sample delay before downsampling. To implement this, we downsample two shifted versions of input image (corresponding to $(\Delta x, \Delta y) = \{(0,0),(1,1)\}$), pass two downsampled images from our directional filter bank, and fuse the outputs to construct one output image per filter in directional filter bank. Fusing method used in thesis is simply taking square of images, summing them, and taking the square root of the sum.

A variant of this multi-scale filtering framework uses four shifted versions instead of two (corresponding to $(\Delta x, \Delta y) = \{(0,0),(1,0),(0,1),(1,1)\}$). Although this increases the accuracy by average 1%, it also doubles the computational complexity. This speed vs. accuracy trade-off should be evaluated for potential applications.

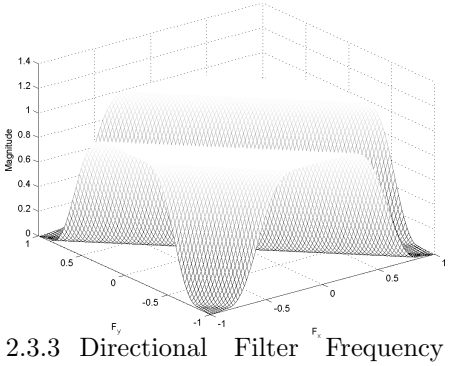
The lowpass filter used in downsampling f_0 can be the corresponding lowpass filter of a wavelet filter bank. If f_0 is chosen as such, or it can be a simple half-band filter. The low-low sub-image can be filtered by directional filters to obtain



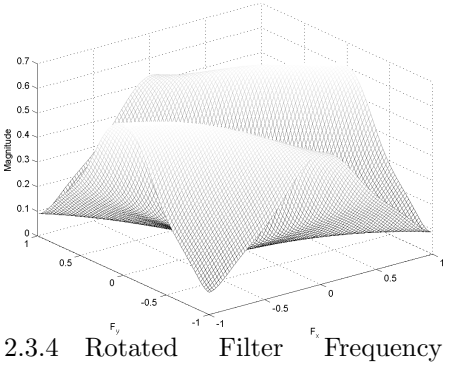
2.3.1 Directional Filter Frequency Response($\theta = -63.43°$)



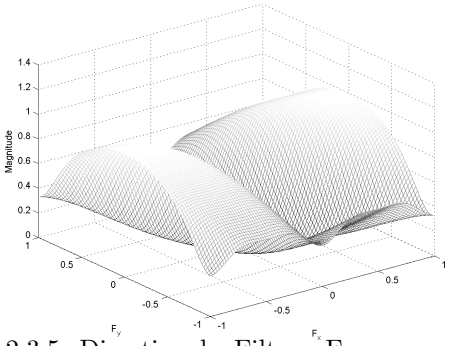
2.3.2 Rotated Filter Frequency Response($\theta = -63.43°$)



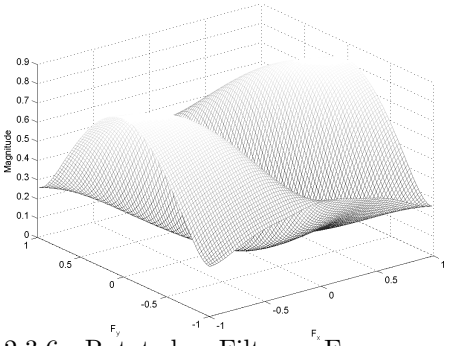
2.3.3 Directional Filter Frequency Response($\theta = -45°$)



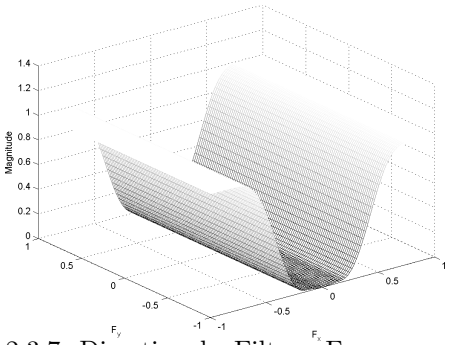
2.3.4 Rotated Filter Frequency Response($\theta = -45°$)



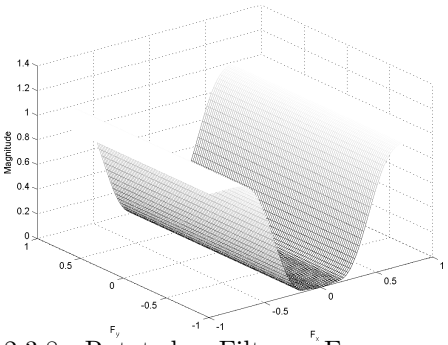
2.3.5 Directional Filter Frequency Response($\theta = -26.56°$)



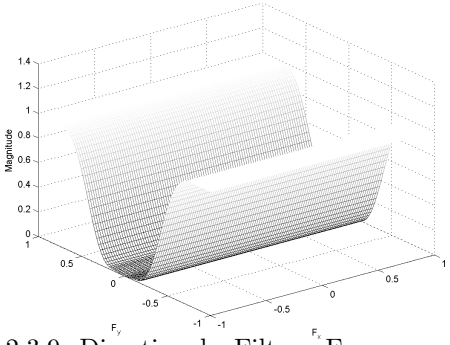
2.3.6 Rotated Filter Frequency Response($\theta = --26.56°$)



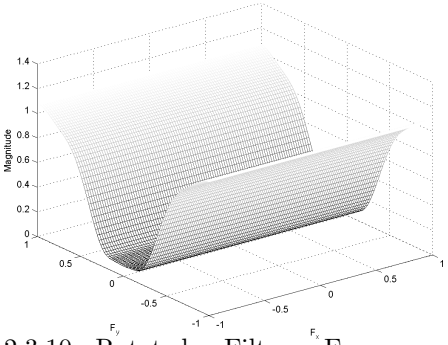
2.3.7 Directional Filter Frequency Response($\theta = 0°$)



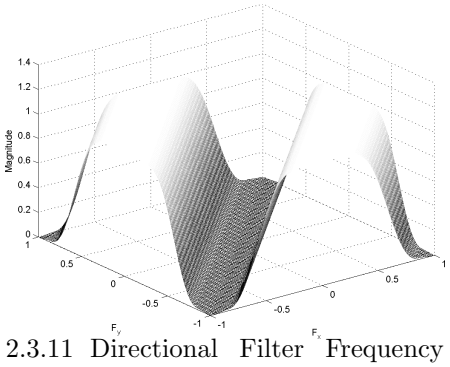
2.3.8 Rotated Filter Frequency Response($\theta = -0°$)



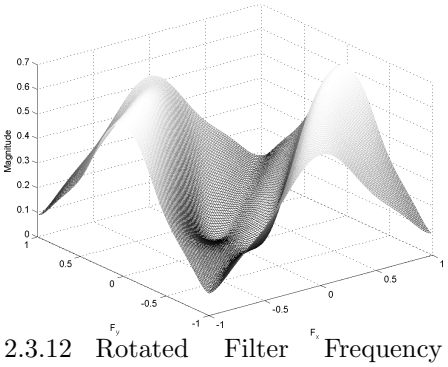
2.3.9 Directional Filter Frequency Response($\theta = 26.56°$)



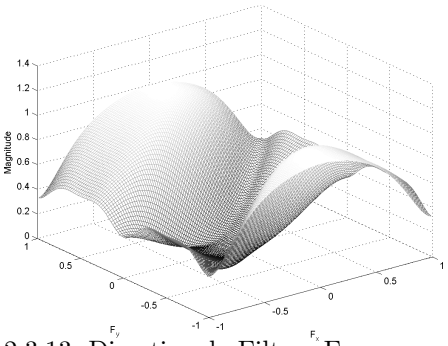
2.3.10 Rotated Filter Frequency Response($\theta = 26.56°$)



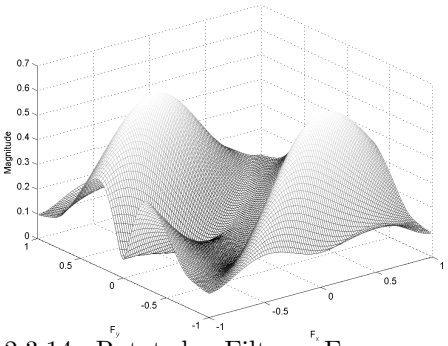
2.3.11 Directional Filter Frequency Response($\theta = 45°$)



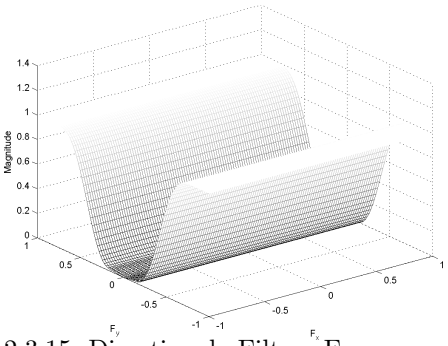
2.3.12 Rotated Filter Frequency Response($\theta = 45°$)



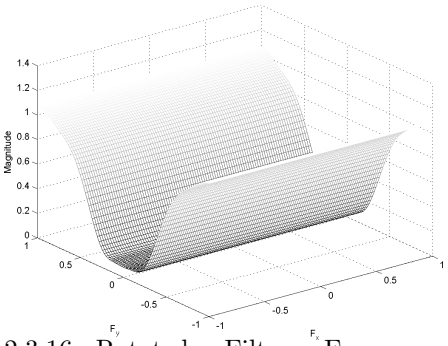
2.3.13 Directional Filter Frequency Response($\theta = 63.43°$)



2.3.14 Rotated Filter Frequency Response($\theta = 63.43°$)

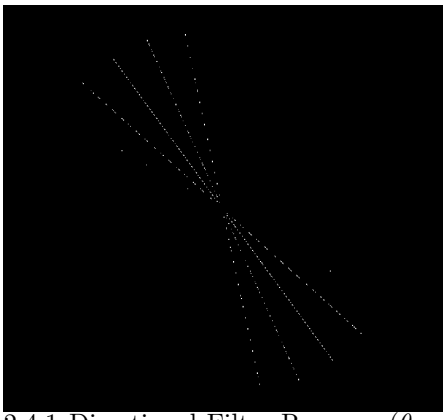


2.3.15 Directional Filter Frequency Response($\theta = 90°$)

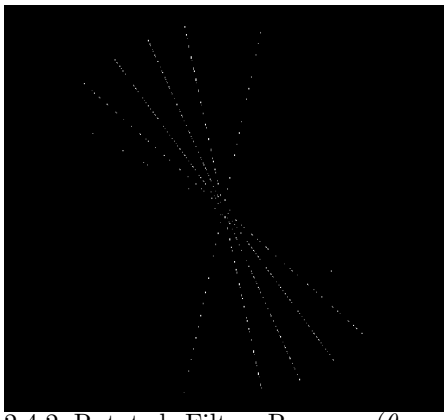


2.3.16 Rotated Filter Frequency Response($\theta = 90°$)

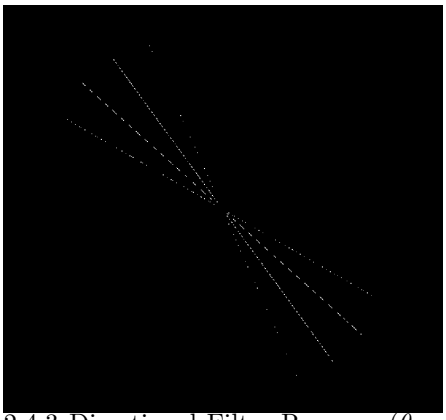
Figure 2.3: Frequency responses of directional and rotated filters, for $\theta = 0°, \pm 26.56°, \pm 45°, \pm 63.43°, 90°$



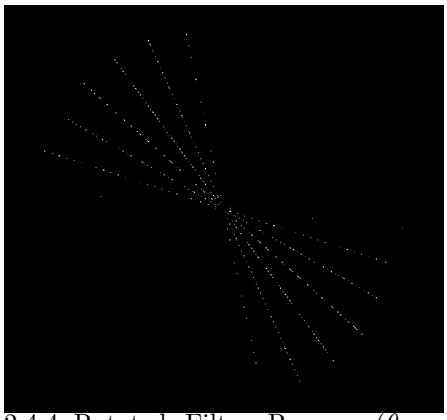
2.4.1 Directional Filter Response($\theta = -63.43°$)



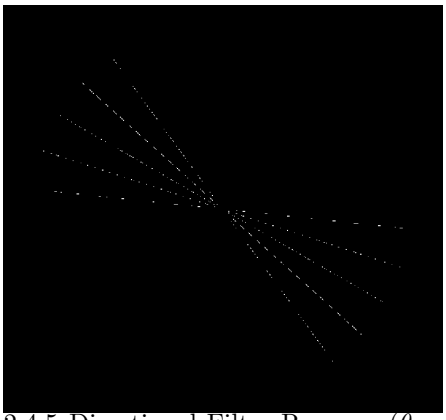
2.4.2 Rotated Filter Response($\theta = -63.43°$)



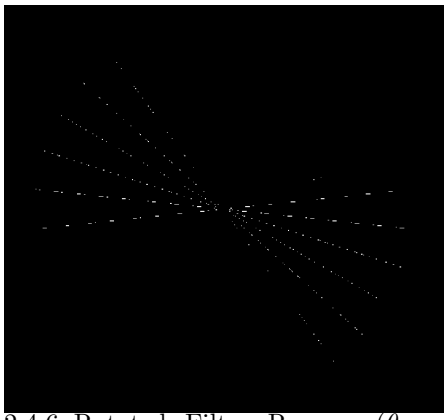
2.4.3 Directional Filter Response($\theta = -45°$)



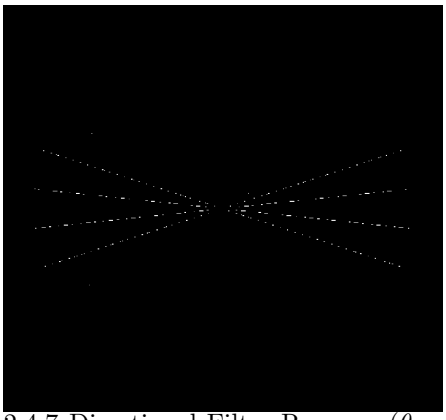
2.4.4 Rotated Filter Response($\theta = -45°$)



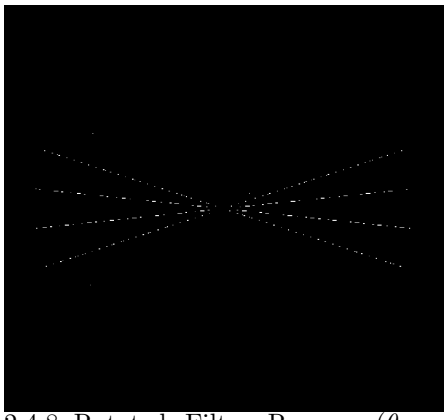
2.4.5 Directional Filter Response($\theta = -26.56°$)



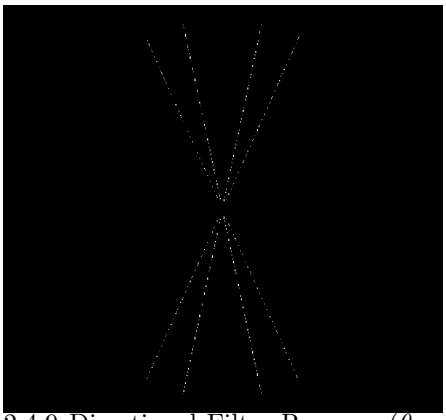
2.4.6 Rotated Filter Response($\theta = --26.56°$)



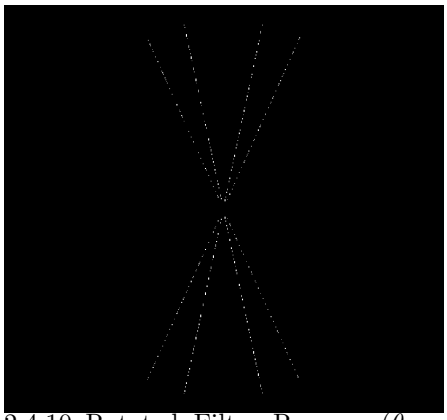
2.4.7 Directional Filter Response($\theta = 0°$)



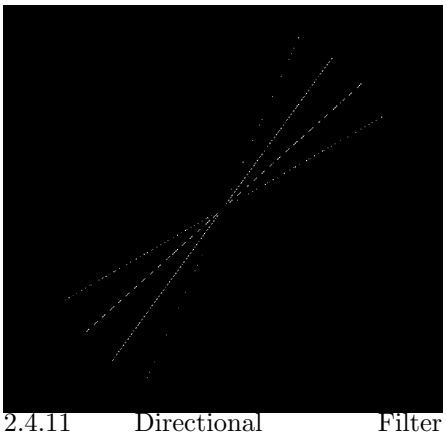
2.4.8 Rotated Filter Response($\theta = -0°$)



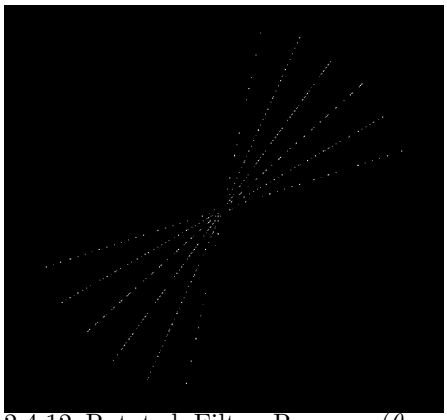
2.4.9 Directional Filter Response($\theta = 26.56°$)



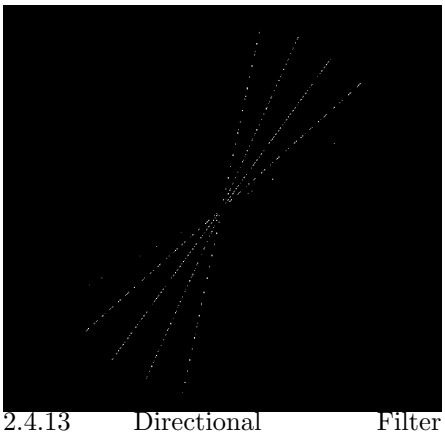
2.4.10 Rotated Filter Response($\theta = 26.56°$)



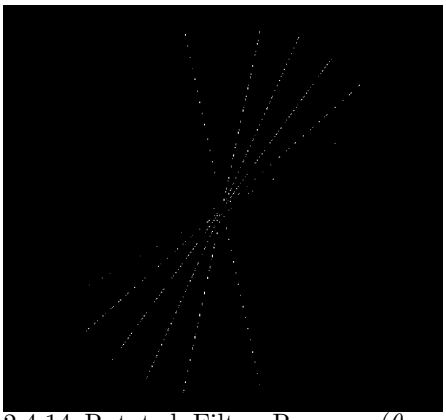
2.4.11 Directional Filter Response($\theta = 45°$)



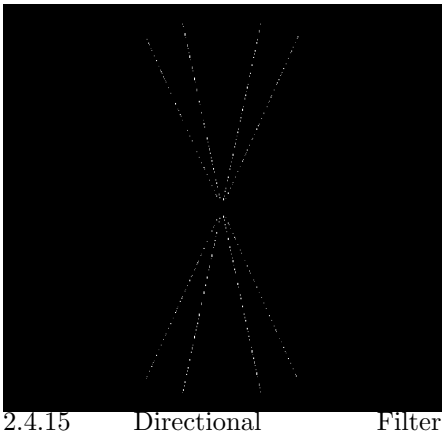
2.4.12 Rotated Filter Response($\theta = 45°$)



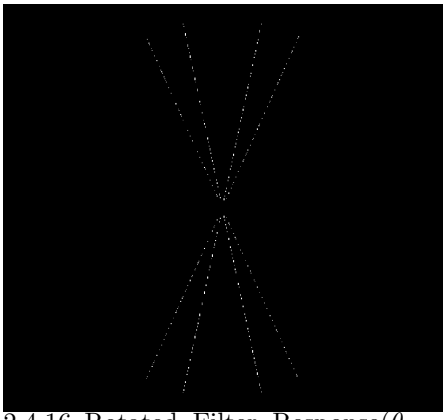
2.4.13 Directional Filter Response($\theta = 63.43°$)



2.4.14 Rotated Filter Response($\theta = 63.43°$)



2.4.15 Directional Filter Response($\theta = 90°$)



2.4.16 Rotated Filter Response($\theta = 90°$)

Figure 2.4: Largest %2 Responses of directional and rotated filters on fireworks pattern, for $\theta = 0°, \pm 26.56°, \pm 45°, \pm 63.43°, 90°$

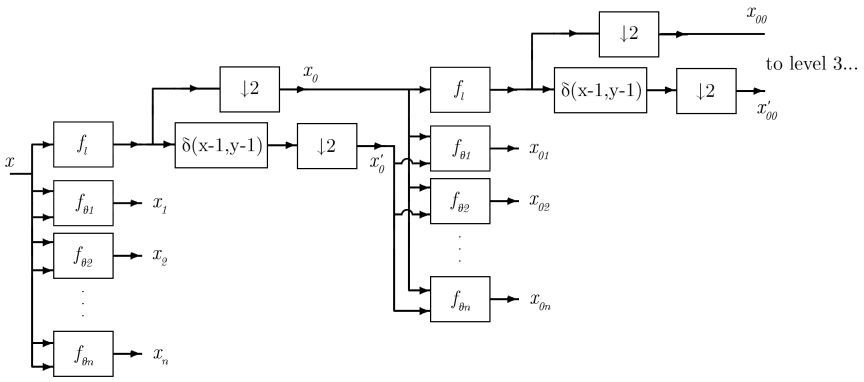


Figure 2.5: Image flowchart directional filtering framework

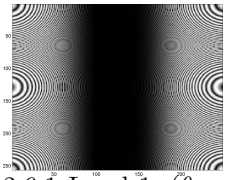
the second level directional subimages and corresponding feature values. This process can be repeated several times depending on the nature of input images. The filtering flow diagram is shown in Figure 2.5.

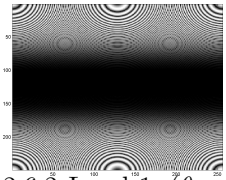
In our experiments, we use two different directional filters, both in 3 scales, $\theta = \{0°, \pm 26.56°, \pm 45°, \pm 63.43°, 90°\}$ and lowpass filter is halfband filter $f_l = [0.25\ 0.5\ 0.25]$. For one filter bank we use Lagrange *à trous* filter [38]: $f_0 = [-0.0625\ 0\ 0.5625\ 1\ 0.5625\ 0\ -0.0625]$ and other uses Kingsburys $8th$ order q-shift analysis filter [39]: $f_0 = [-0.0808\ 0\ 0.4155\ -0.5376\ 0.1653\ 0.0624\ 0\ -0.0248]$

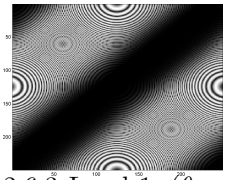
2.2 Complex Wavelet Transform

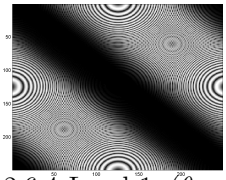
Complex wavelet transform(CWT) is developed by Kingsbury as an improvement over shortcomings of the discrete wavelet transform (DWT). DWT for 2-dimensional (2-D) signals can be implemented using two stages of filtering at each level: first, rows of the image are filtered to generate a pair of horizontal lowpass and highpass images, and then columns of these images are filtered to produce four sub-images.

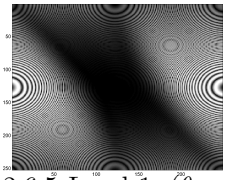
However, as [40] states, there are two fundamental problems with real DWT that makes it unfeasible to model textural features with. First one is the shift

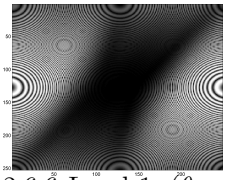

2.6.1 Level 1, ($\theta = 0°$)

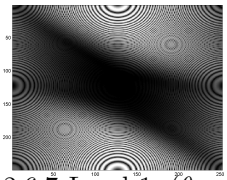

2.6.2 Level 1, ($\theta = 90°$)

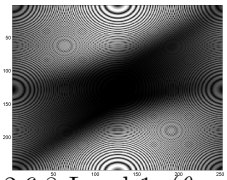

2.6.3 Level 1, ($\theta = 45°$)

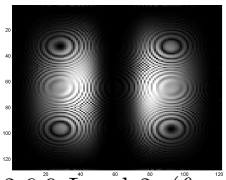

2.6.4 Level 1, ($\theta = -45°$)

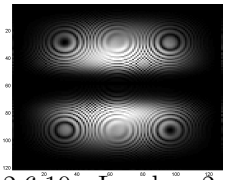

2.6.5 Level 1, ($\theta = -63.43°$)

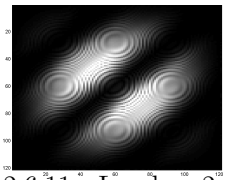

2.6.6 Level 1, ($\theta = 63.43°$)

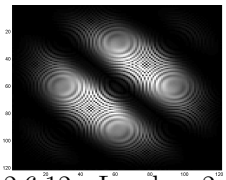

2.6.7 Level 1, ($\theta = -26.56°$)

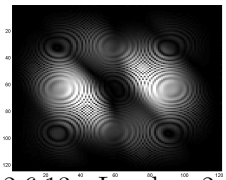

2.6.8 Level 1, ($\theta = 26.56°$)

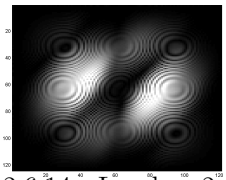

2.6.9 Level 2, ($\theta = -0°$)

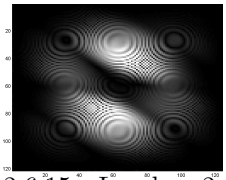

2.6.10 Level 2, ($\theta = 90°$)

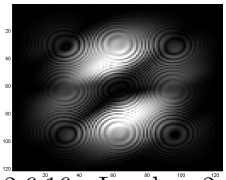

2.6.11 Level 2, ($\theta = 45°$)

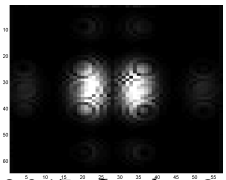

2.6.12 Level 2, ($\theta = -45°$)

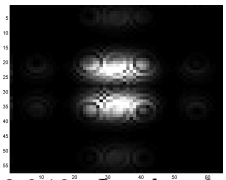

2.6.13 Level 2, ($\theta = -63.43°$)

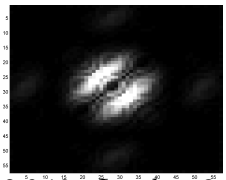

2.6.14 Level 2, ($\theta = 63.43°$)

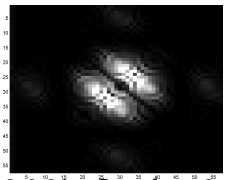

2.6.15 Level 2, ($\theta = -26.56°$)

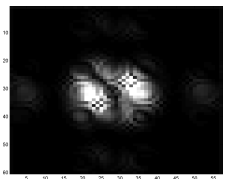

2.6.16 Level 2, ($\theta = 26.56°$)

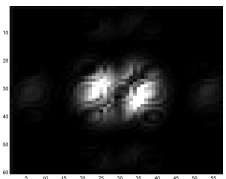

2.6.17 Level 3, ($\theta = 0°$)

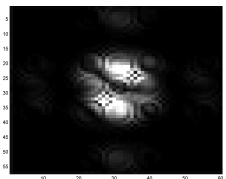

2.6.18 Level 3, ($\theta = 90°$)


2.6.19 Level 3, ($\theta = 45°$)


2.6.20 Level 3, ($\theta = -45°$)


2.6.21 Level 3, ($\theta = -63.43°$)


2.6.22 Level 3, ($\theta = 63.43°$)


2.6.23 Level 3, ($\theta = -26.56°$)

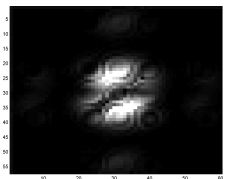

2.6.24 Level 3, ($\theta = 26.56°$)

Figure 2.6: Directional Filter Responses of zoneplate pattern

variance. Note that implementation of DWT is maximally decimated, total output sample size is equal to the input size for each level due to decimation at each level. This provides zero redundancy, however it also causes aliasing and results in shift variance. The other problem is poor directional selectivity. Since both highpass row filters and highpass column filters in DWT are real, their spectrum are symmetric. Therefore Lo-Hi and Hi-Lo filters select high frequencies in both positive and negative directions in their corresponding orientations (Hi-Lo filters select horizontal frequencies, Lo-Hi filters select vertical frequencies). Hi-Hi filter however, selects diagonal edges with both positive and negative gradients, due to having passbands in all four corners of the frequency plane. Output of Hi-Hi filter cannot separate $+45°$ diagonal edges from $-45°$ diagonal edges; which causes poor directional selectivity [41]. This phenomenon can be seen clearly from the impulse responses in figure 2.7

As a solution, Kingsbury in [40] proposes dual-tree complex wavelet transform (DT-CWT), which is almost shift invariant, directionally selective, has perfect reconstruction, introduces minimal redundancy (4:1 for 2D signals) and achieves order-N computation. [40]

Filter selection is a critical issue and there are many papers on designing filters with suitable properties for wavelet transform. Following 6-tap filter designed in [39] and Farras filters [42] are used in our implementation:

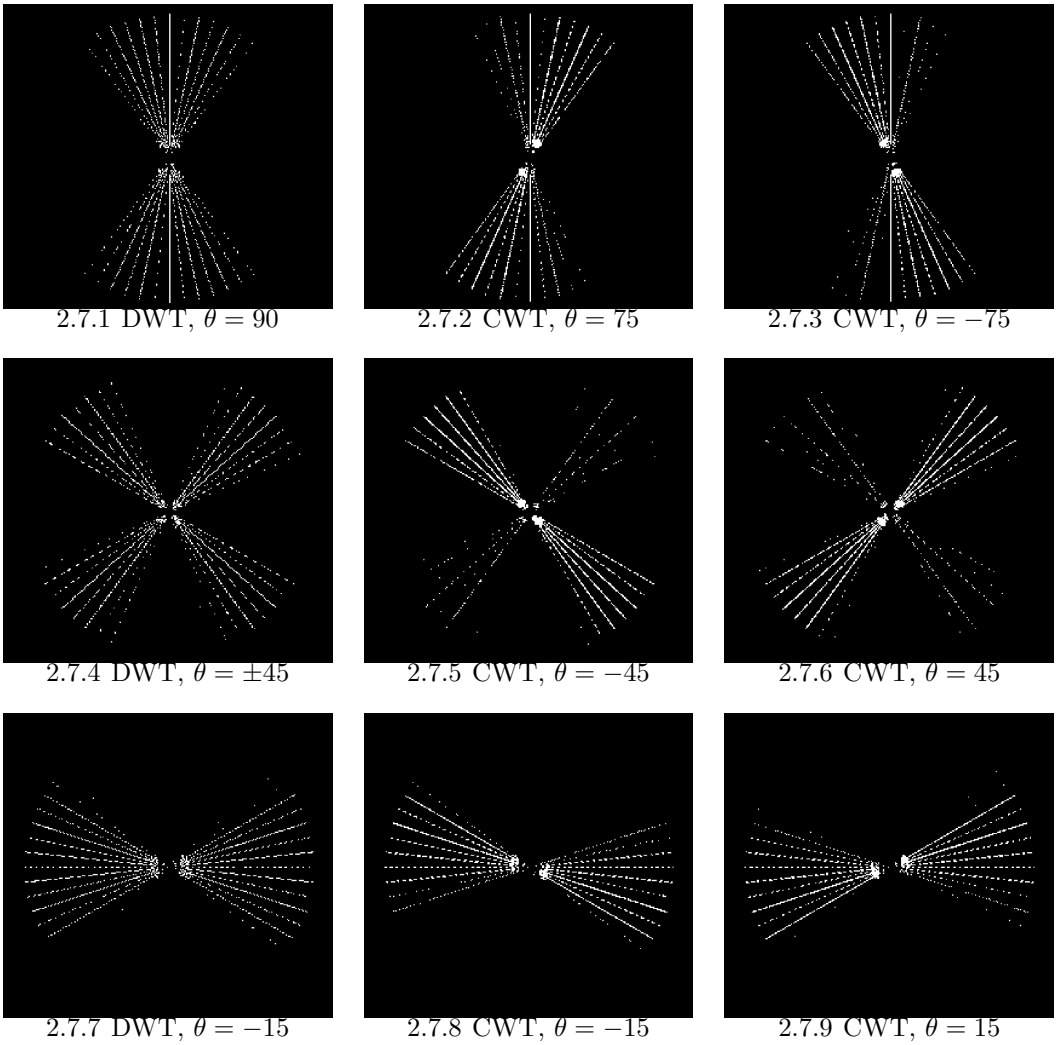
$$Lo_p[n] = [0, -0.0884, 0.0884, 0.6959, 0.6959, 0.0884, -0.0884, 0.0112, 0.0112] \tag{2.5}$$

$$Hi_p[n] = [0.0112, 0.0112, -0.0884, 0.0884, 0.6959, 0.6959, 0.0884, -0.0884, 0, 0] \tag{2.6}$$

$$Lo[n] = [0.0351, 0, -0.0883, 0.2339, 0.7603, 0.5875, 0, -0.1143, 0, 0] \tag{2.7}$$

$$Hi[n] = [0, 0, -0.1143, 0, 0.5875, 0.7603, 0.2339, -0.0883, 0, 0.0351] \tag{2.8}$$

Note that Equations 2.5-2.8 only shows the real parts of the filters. Imaginary



2.7.1 DWT, $\theta = 90$ 2.7.2 CWT, $\theta = 75$ 2.7.3 CWT, $\theta = -75$

2.7.4 DWT, $\theta = \pm 45$ 2.7.5 CWT, $\theta = -45$ 2.7.6 CWT, $\theta = 45$

2.7.7 DWT, $\theta = -15$ 2.7.8 CWT, $\theta = -15$ 2.7.9 CWT, $\theta = 15$

Figure 2.7: Top %2 responses of DWT (left column) and CWT(center and right columns) to fireworks pattern for at level 1

parts of the filters can we calculated by reversing and taking the negative of each respective filter.

Since DT-CWT produces output images with different size at each tree level due to decimation, and these sizes depend on input image size, it is not feasible to use output images of DT-CWT directly. Instead we are using mean and variance of the absolute value of outputs of 3 level complex wavelet tree. We have tried several levels, and found that the recognition rate does not increase noticeably after 3rd level. We have also tried taking mean and variance of real and imaginary parts of output images separately and concatenating them, but it gave poorer performance than the absolute values. Overall, our feature vector includes means and variances of 18 images (6 outputs for each level), resulting in a 36-element feature vector[1].

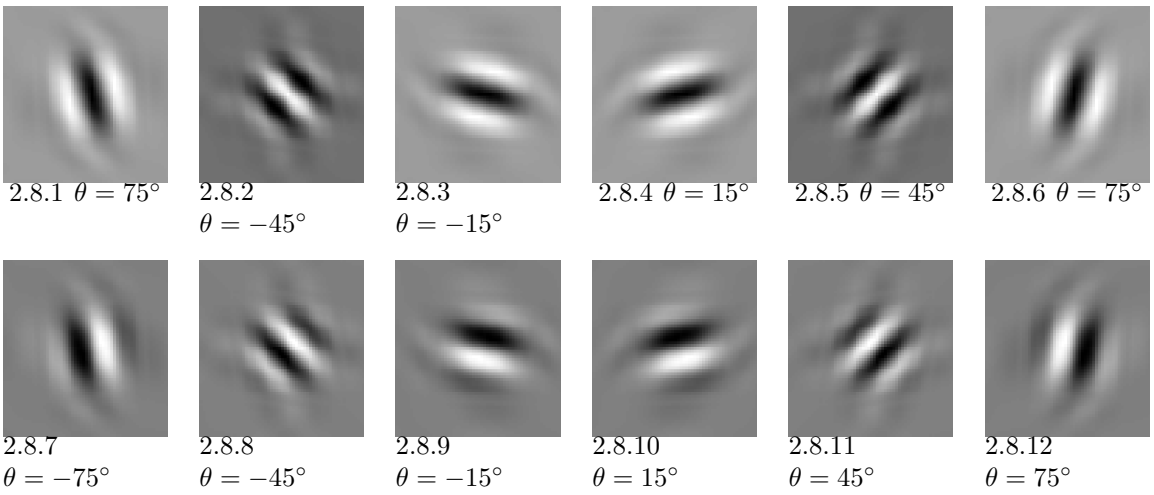
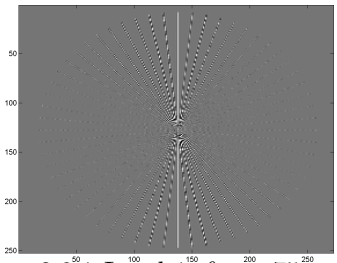


Figure 2.8: 2D Impulse responses of CWT at level 4(top row:real, bottom row:imaginary)

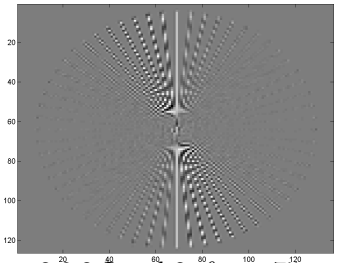
2.3 Curvelet Transform

The problem in Figure 2.10 is the main motivation behind curvelets. Curvelets are first introduced in 1999 [43] and revised in 2003 [44], both by Candes and

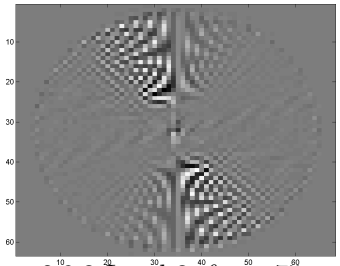
[1]DT-CWT implementation given in http://eeweb.poly.edu/iselesni/WaveletSoftware/index.html is used in this thesis.



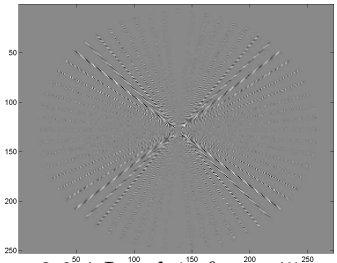
2.9.1 Level 1, $\theta = -75$



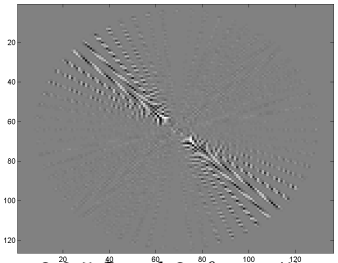
2.9.2 Level 2, $\theta = -75$



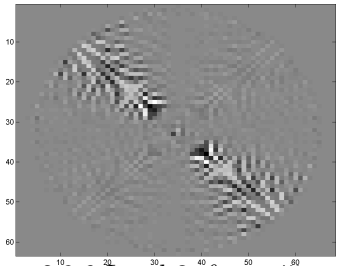
2.9.3 Level 3, $\theta = -75$



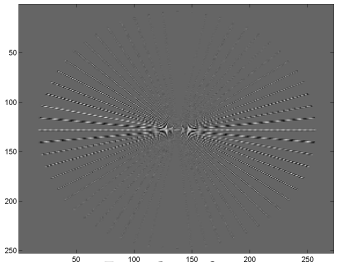
2.9.4 Level 1, $\theta = -45$



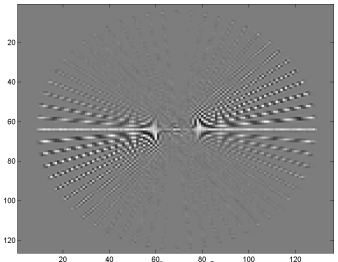
2.9.5 Level 2, $\theta = -45$



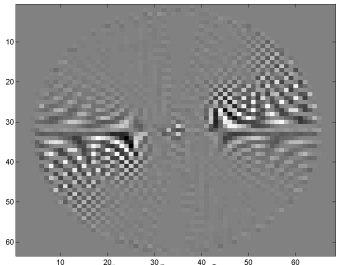
2.9.6 Level 3, $\theta = -45$



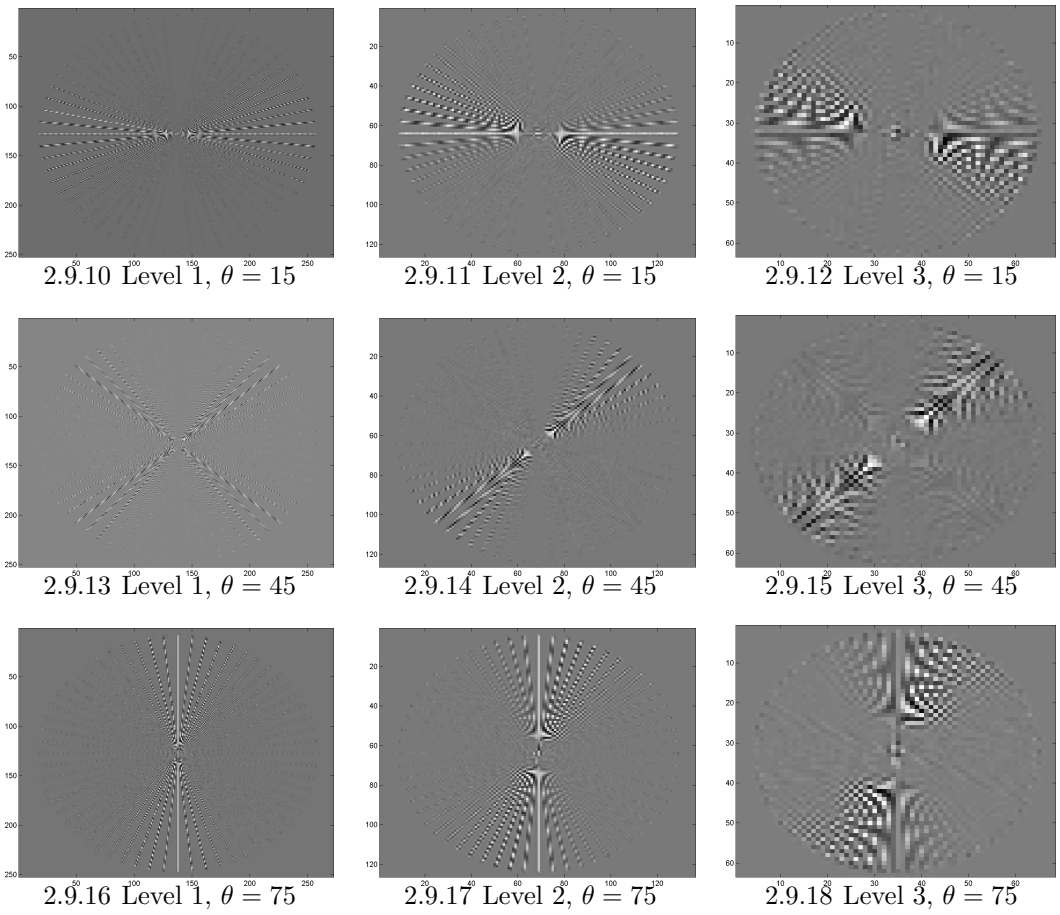
2.9.7 Level 1, $\theta = -15$



2.9.8 Level 2, $\theta = -15$



2.9.9 Level 3, $\theta = -15$

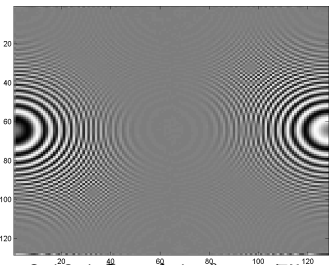


2.9.10 Level 1, $\theta = 15^{\circ}$ 2.9.11 Level 2, $\theta = 15^{\circ}$ 2.9.12 Level 3, $\theta = 15^{\circ}$

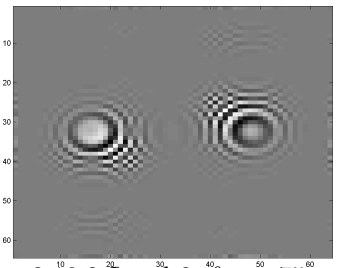
2.9.13 Level 1, $\theta = 45^{\circ}$ 2.9.14 Level 2, $\theta = 45^{\circ}$ 2.9.15 Level 3, $\theta = 45^{\circ}$

2.9.16 Level 1, $\theta = 75^{\circ}$ 2.9.17 Level 2, $\theta = 75^{\circ}$ 2.9.18 Level 3, $\theta = 75^{\circ}$

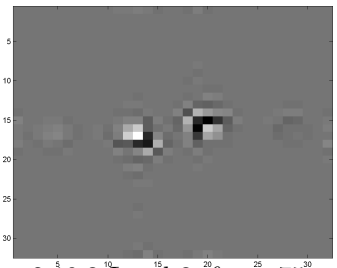
Figure 2.9: Complex Wavelet Transform of the fireworks pattern, for 3 scales and $\theta = \pm 15^{\circ}, \pm 45^{\circ}, \pm 75^{\circ}$



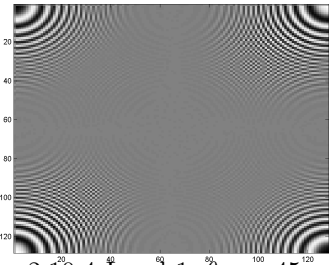
2.10.1 Level 1, $\theta = -75$



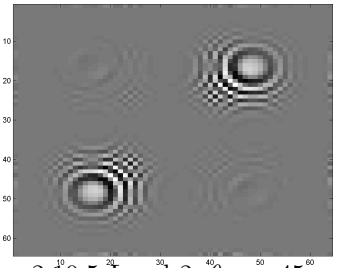
2.10.2 Level 2, $\theta = -75$



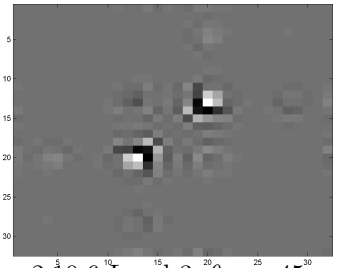
2.10.3 Level 3, $\theta = -75$



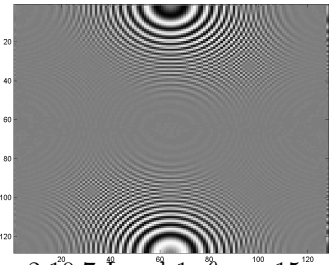
2.10.4 Level 1, $\theta = -45$



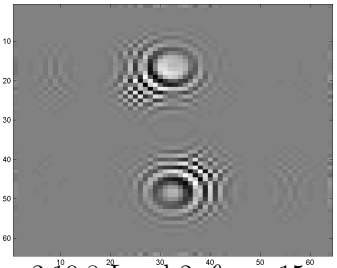
2.10.5 Level 2, $\theta = -45$



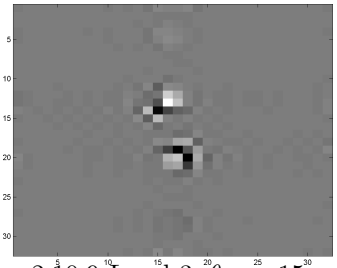
2.10.6 Level 3, $\theta = -45$



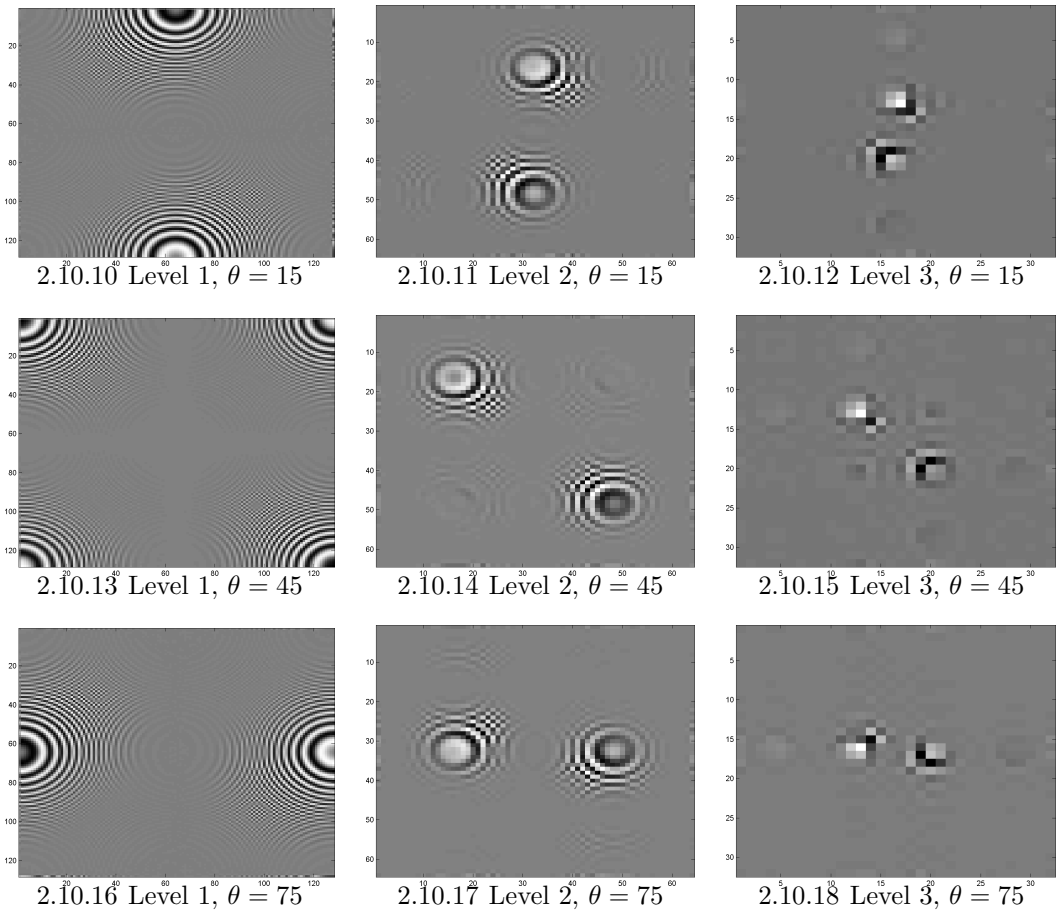
2.10.7 Level 1, $\theta = -15$



2.10.8 Level 2, $\theta = -15$



2.10.9 Level 3, $\theta = -15$



2.10.10 Level 1, $\theta = 15^\circ$ 2.10.11 Level 2, $\theta = 15^\circ$ 2.10.12 Level 3, $\theta = 15^\circ$

2.10.13 Level 1, $\theta = 45^\circ$ 2.10.14 Level 2, $\theta = 45^\circ$ 2.10.15 Level 3, $\theta = 45^\circ$

2.10.16 Level 1, $\theta = 75^\circ$ 2.10.17 Level 2, $\theta = 75^\circ$ 2.10.18 Level 3, $\theta = 75^\circ$

Figure 2.10: Complex Wavelet Transform of the zoneplate pattern, for 3 scales and $\theta = \pm 15^\circ, \pm 45^\circ, \pm 75^\circ$

Donoho. The original curvelets are obtained by filtering an input image into bandpass images then applying windowed Ridgelet transform [45] in each bandpass image. The motivation for this is that by smooth windowing, segments of smooth curves with would look straight in subimages, hence they can be captured by a local ridgelet transform (Ridgelets are designed for efficiently capturing straight lines.) Curvelet transform is an improvement over wavelets, due to some of its properties which regular (or complex) wavelets are lacking. These are optimally sparse representation of object with edges, optimal sparse representation of wave propagators, and optimal image reconstruction in severely ill-posed problems.

Revised version of the curvelets does not utilize ridgelets. First, the frequency space is divided into dyadic annuli based on concentric squares, and each annulus is divided into trapezoidal regions. The redesigned version can be implemented by two different ways: Unequispaced FFTs or wrapping. Although each method calculate transform coefficients via different ways, computational complexity is $\mathcal{O}(n^2 \log n)$ for $n \times n$ image for both methods. In this thesis wrapping method is used. A curvelet atom with scale s, orientation $\theta \in [0, \pi)$, position $\mathbf{y} \in [0, 1]^2$ is defined as

$$\phi_{s,\mathbf{y},\theta}(\mathbf{x}) = \phi_s(R_0^{-1}(\mathbf{x} - \mathbf{y})), \tag{2.9}$$

where $\phi_s(\mathbf{x}) \approx s^{-3/4}\phi(s^{-1/2}x_1, s^{-1}x_2)$ is approximately a parabolic stretch of a curvelet function ϕ with vanishing moments in vertical direction. At scale s, a curvelet atom is thus a needle oriented in the direction θ whose envelope is a specified ridge of effective length $s^{1/2}$ and width s, and which displays an oscillatory behavior transverse to the ridge. Hence, the curvelet atoms benefit from anisotropic (more specifically, parabolic) scaling property $width = length^2$ which is a major departure from oriented wavelets. It follows from this property in frequency domain that, a curvelet atom with width 2^{-2j} have 2^j orientations, that is $\#orientations = \frac{1}{\sqrt{scale}}$. It is clear to see from this relation that number of orientations double at every other scale, which gives curvelets exceptional directional selectivity. The curvelet transform of fireworks pattern and zoneplate pattern can be seen in Figures 2.12 and 2.13, Respectively. When compared with CWT of

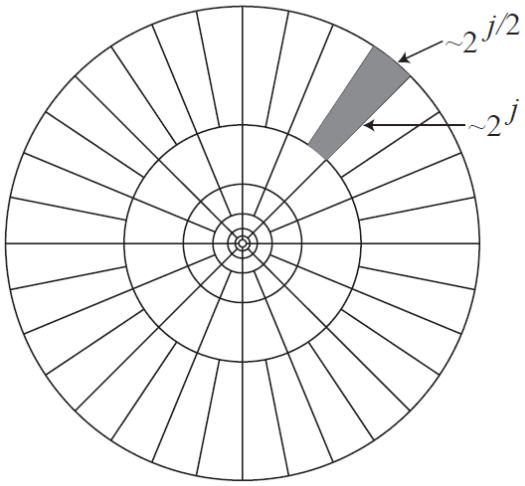


Figure 2.11: Spectral Decomposition of the curvelet transform, in frequency domain. Notice the anisotropic scaling in frequency plane, and increasing angular resolution in finer scales. angular resolution doubles every other scale.

zoneplate pattern (Figure 2.10, it is clear that curvelets can capture edges with high accuracy.

2.4 Contourlet Transform

Contourlets are developed by Do and Vetterli [1] after the success of Curvelets. Contourlets can be considered as a low-redundancy discrete approximation of curvelets. They are designed in the spatial domain (instead of frequency plane as in curvelets) aiming at a close-to-critical directional representation. Their construction is based on Laplacian pyramid. The low pass part of the pyramid is further decomposed with. Each difference image obtained from the pyramid is subject to directional filter banks. A contourlet decompositions of fireworks and zoneplate patterns are illustrated in Figure 2.15 and 2.16. The contourlet transform is overcomplete by a factor of 4/3, which is inherited from the pyramidal framework. Its approximation rate is similar to that of curvelets.

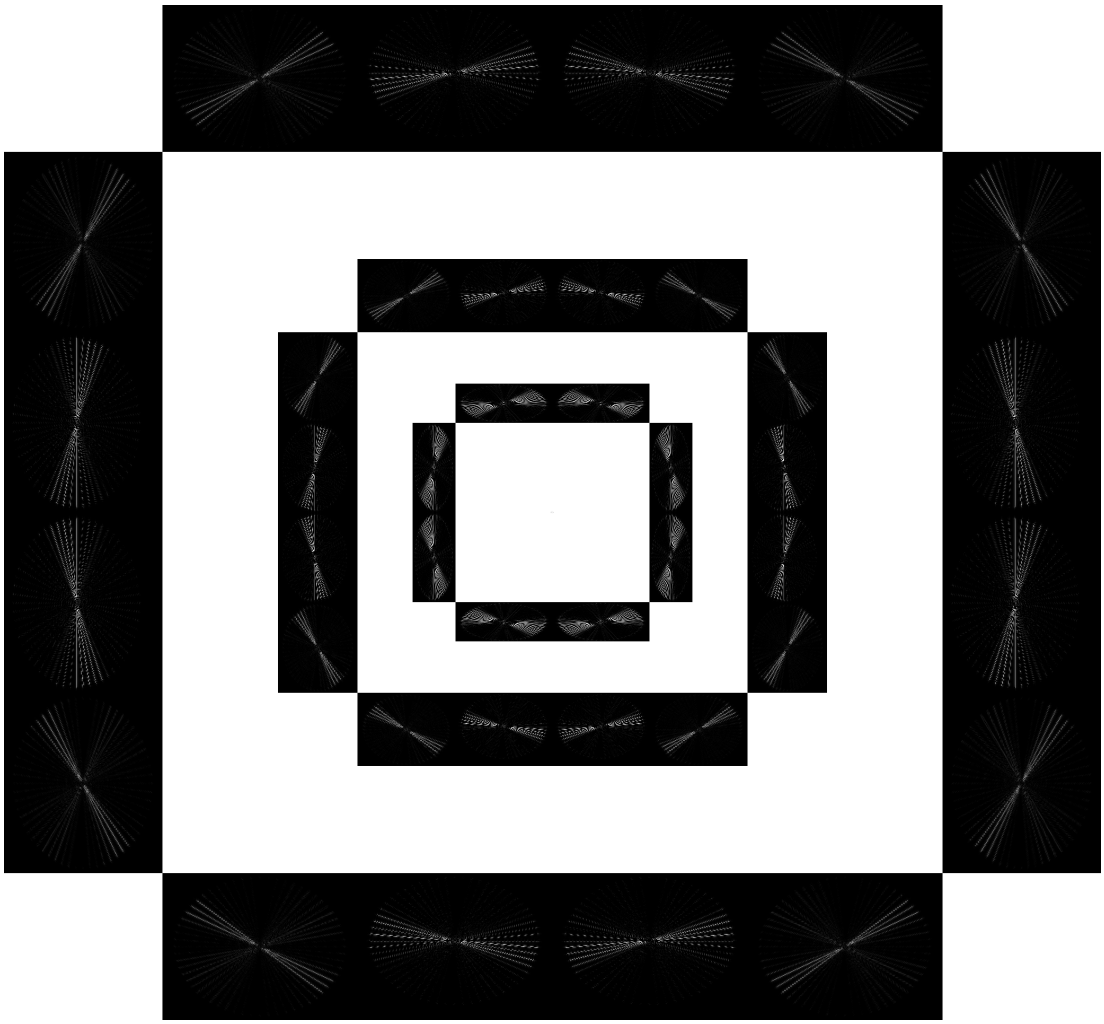


Figure 2.12: Curvelet Transform of the fireworks pattern, for 3 scales

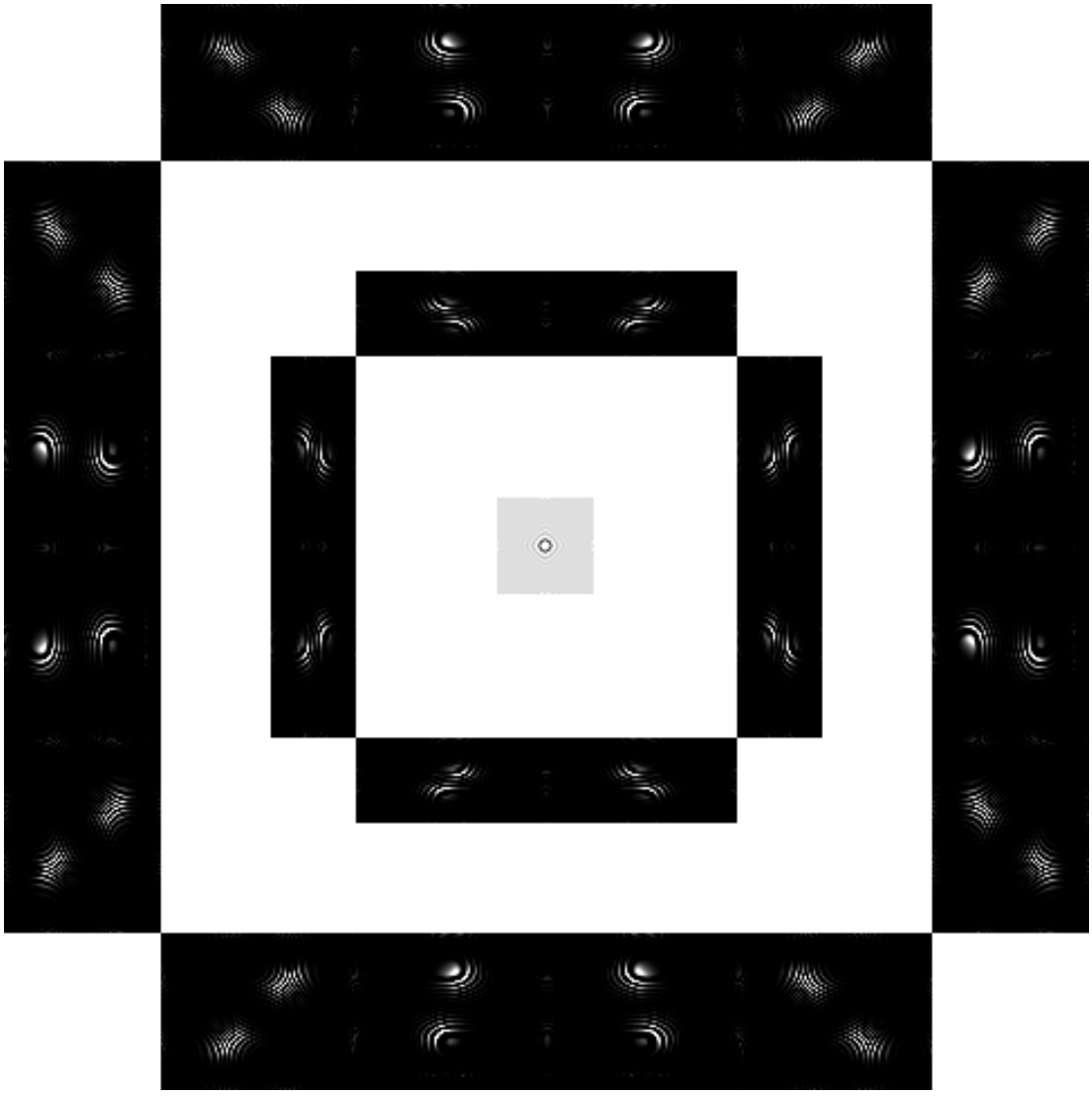


Figure 2.13: Curvelet Transform of the zoneplate pattern, for 3 scales

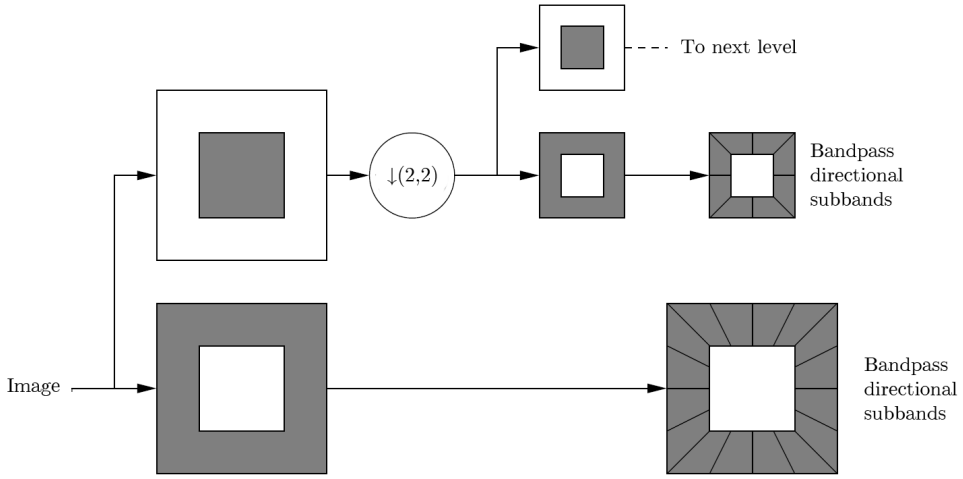


Figure 2.14: Signal flow diagram for one level of the contourlet transform.(Taken from [1])

We use PKVA filters designed in [46] for both pyramid and directional filtering:

$$g[n] = [-0.0144, 0.0272, -0.0526, 0.0972, -0.1930, \\ 0.6300, 0.6300, -0.1930, 0.0972, -0.0526, 0.0272, -0.0144]$$

2.5 Steerable Pyramids

Steerability theory is a very powerful theory that states, if a filter satisfies certain conditions, response of this filter along any orientation can be calculated by linear combination of output of row filtering and and column filtering with the same filter. In order to calculate filter response along several orientations, one simply calculates basis filter responses once then changes weights of combination to get the responses at each direction [47]. This reduces the computational workload substantially.

Steerable filters can be used to form a pyramid decomposition, hence the term

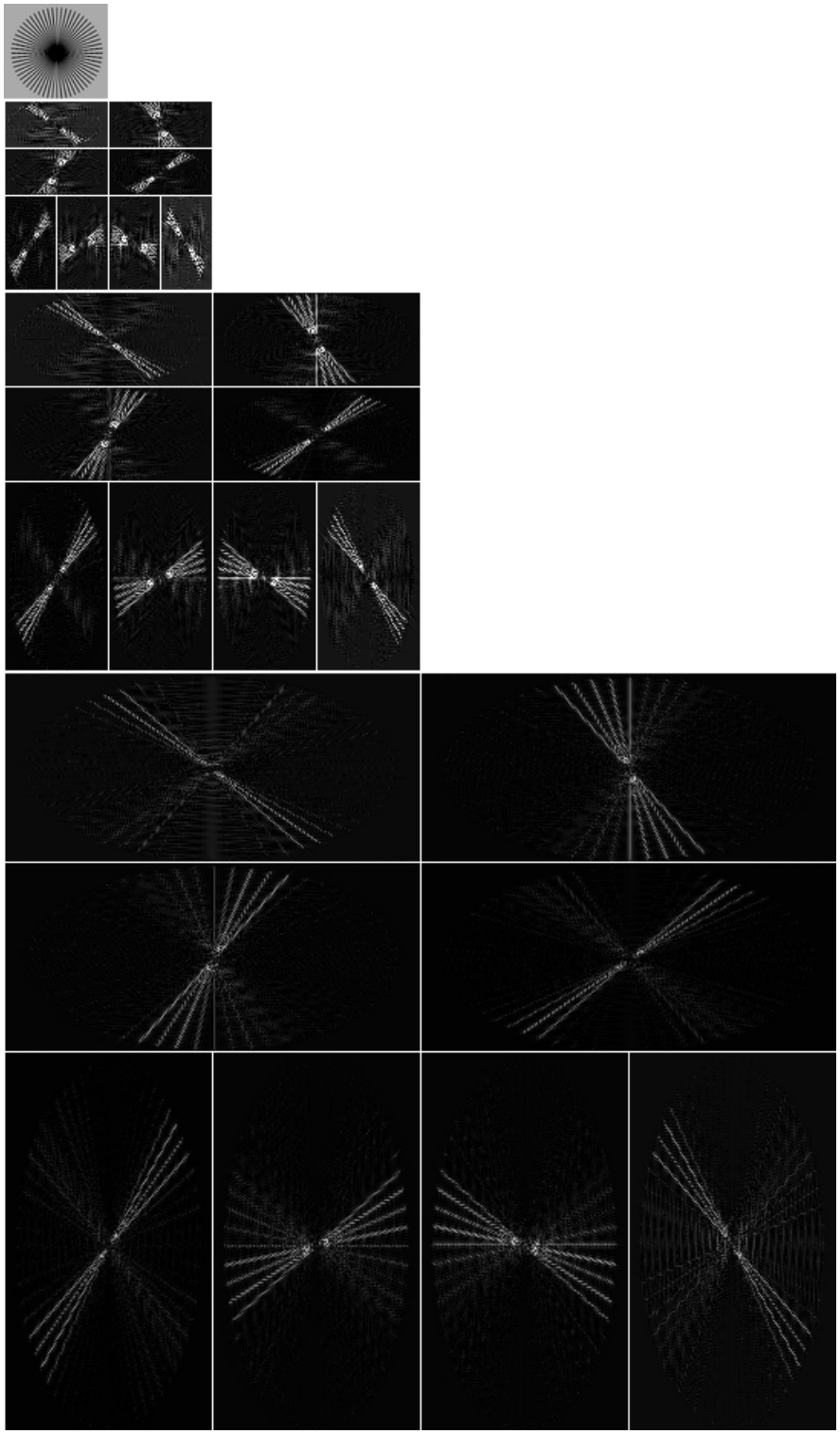


Figure 2.15: Curvelet Transform of the fireworks pattern, for 3 scales

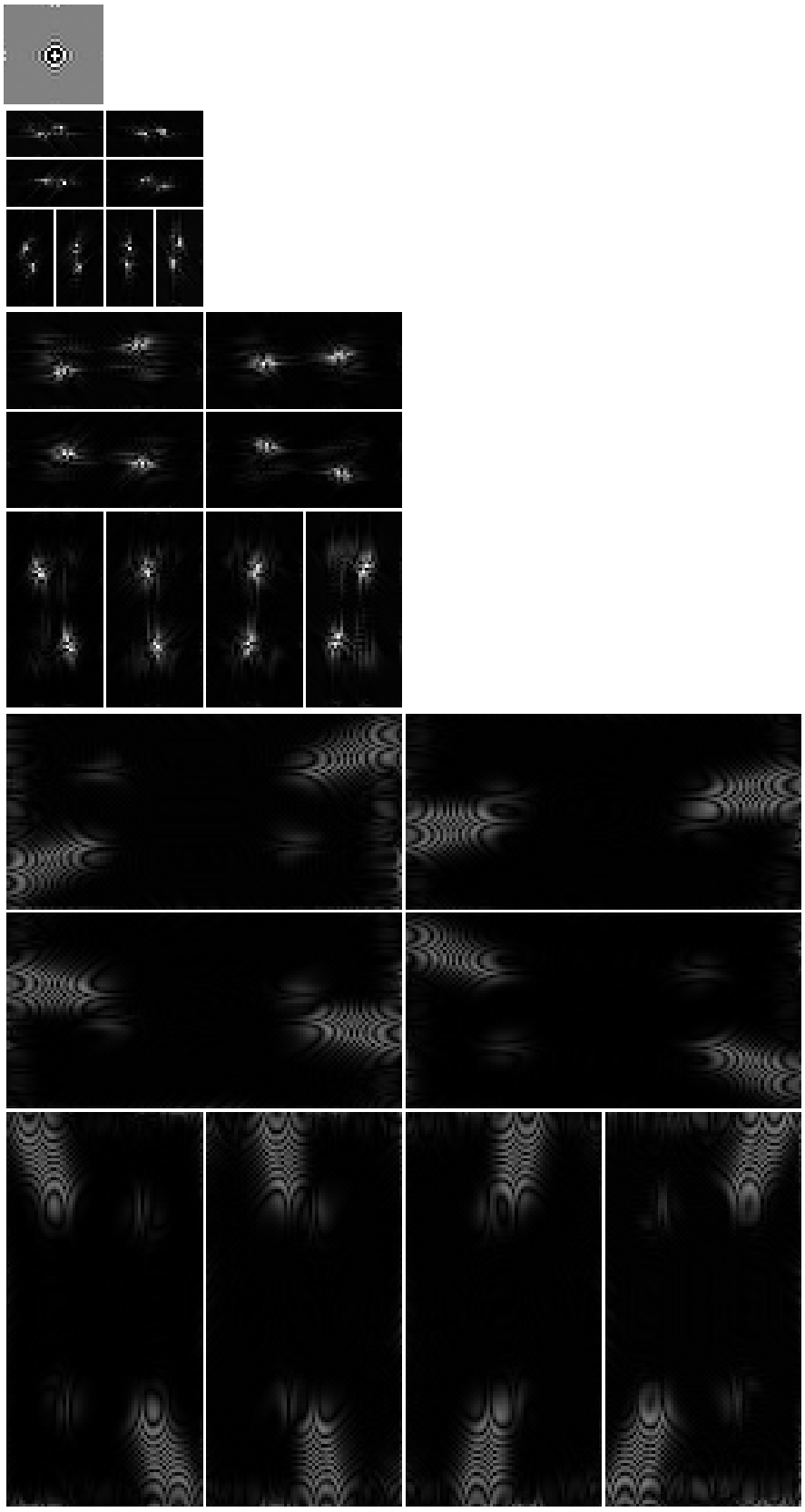


Figure 2.16: Curvelet Transform of the zoneplate pattern, for 3 scales

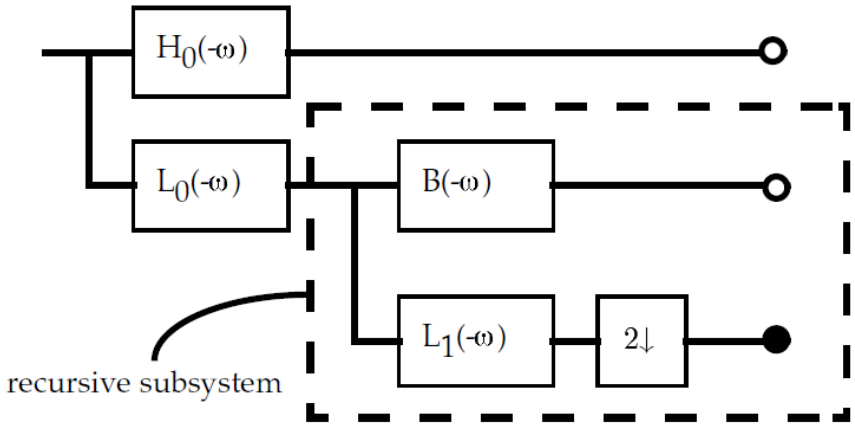


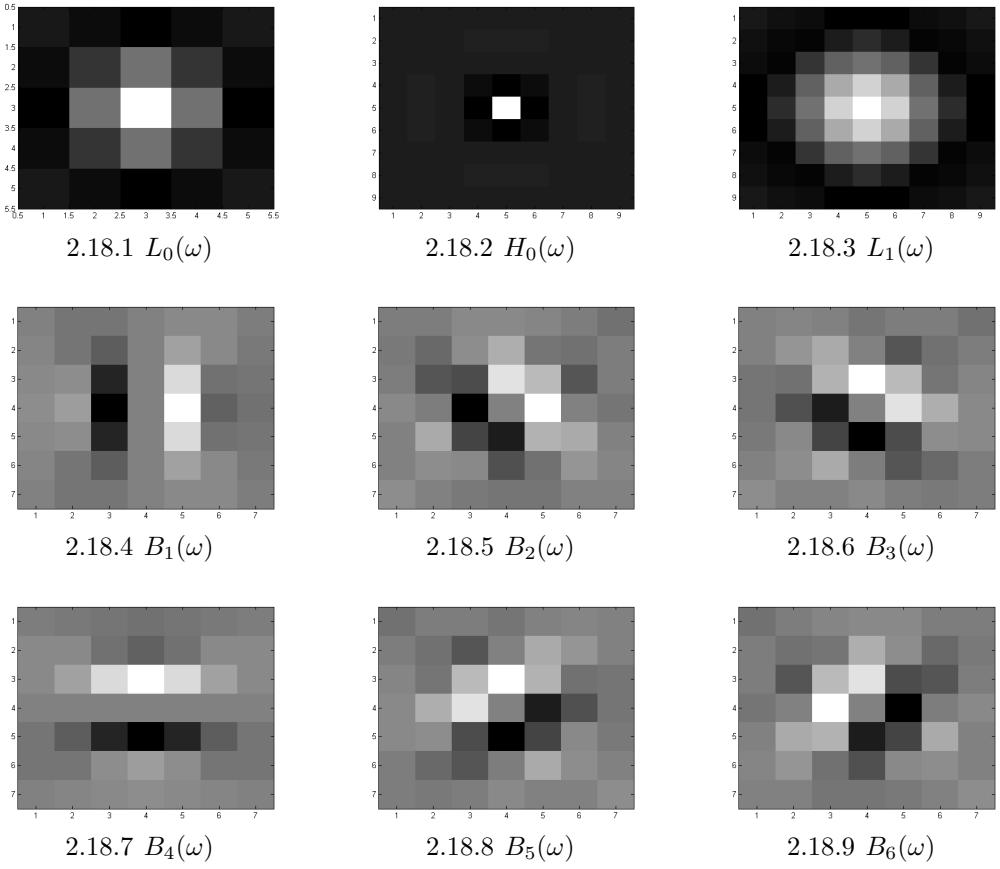
Figure 2.17: System diagram for radial decomposition of steerable pyramid.

'steerable pyramids'. Working principle of steerable pyramids is divided into 2 parts: radial decomposition and angular decomposition. Angular decomposition uses the aforementioned steerable filter scheme and radial decomposition uses pyramid scheme. Pyramidal scheme for radial decomposition can be seen in Figure 2.5. Hollow circles correspond to transform coefficients, and filled circle correspond to the recursive block enclosed by dashed lines.

Pyramid scheme introduces its own constraints for filters in Figure 2.5 in addition to the steerability criteria. Method for designing filters satisfying these criteria is given in [48]. 'sp5' filter set designed with this method is used in this thesis.

When all constraints are satisfied, steerable pyramids are self-inverting, have no aliasing in subbands, and have flexible rotated orientation bands and perfect reconstruction ability, which makes it superior to other multi-scale representations such as Laplacian pyramids and dyadic quadratic mirror filters/wavelets [8]. As a trade-off, it is overcomplete by a factor of $4k/3$ (where k is the number of orientation bands), which is highly redundant.

Steerable pyramid implementation of [8] are used in this thesis. Steerable pyramid outputs for fireworks and zoneplate patterns are presented in Figure 2.19 and 2.20.



2.18.1 $L_0(\omega)$ 2.18.2 $H_0(\omega)$ 2.18.3 $L_1(\omega)$

2.18.4 $B_1(\omega)$ 2.18.5 $B_2(\omega)$ 2.18.6 $B_3(\omega)$

2.18.7 $B_4(\omega)$ 2.18.8 $B_5(\omega)$ 2.18.9 $B_6(\omega)$

Figure 2.18: Filter set used for scale and angular decomposition in steerable pyramid scheme

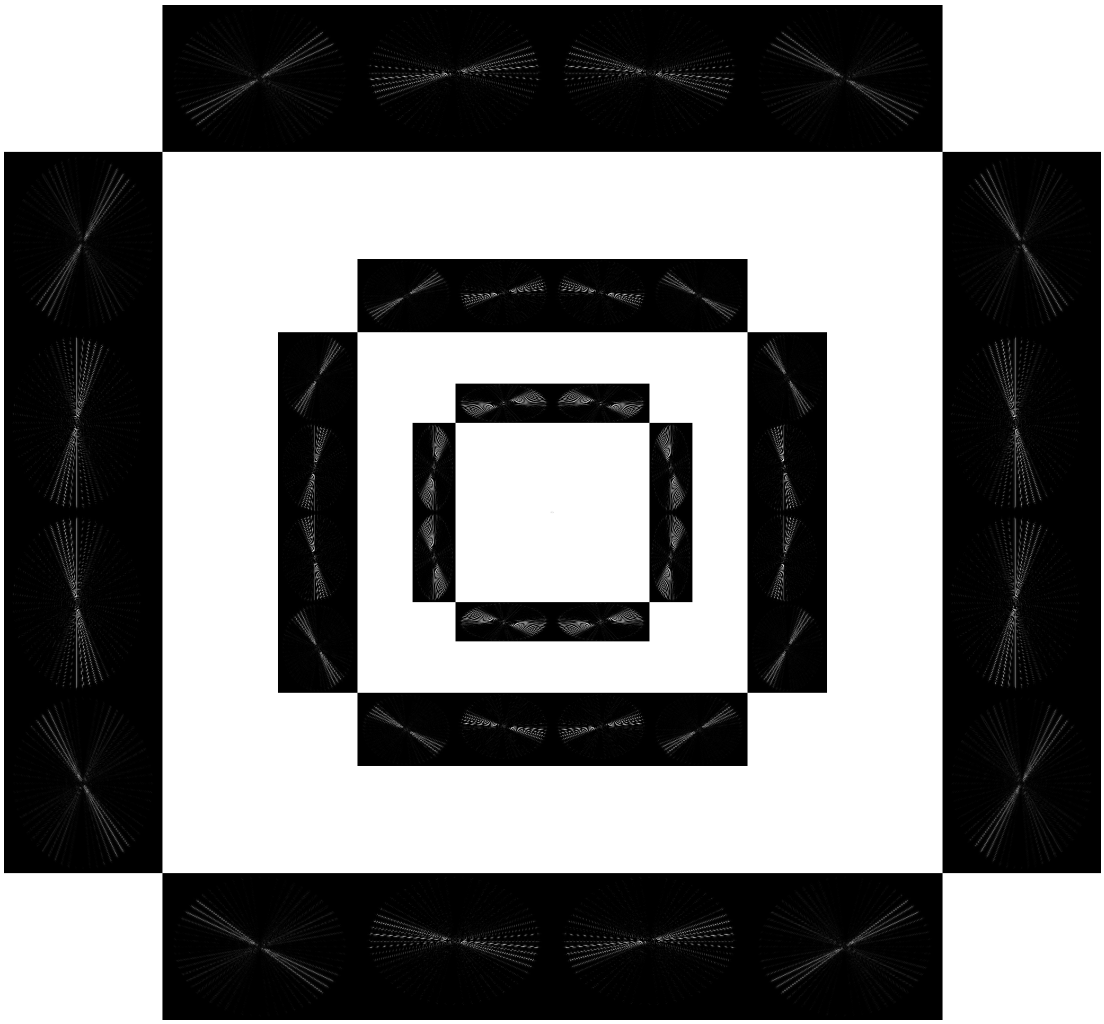


Figure 2.19: Steerable pyramid outputs of the fireworks pattern, for 3 scales

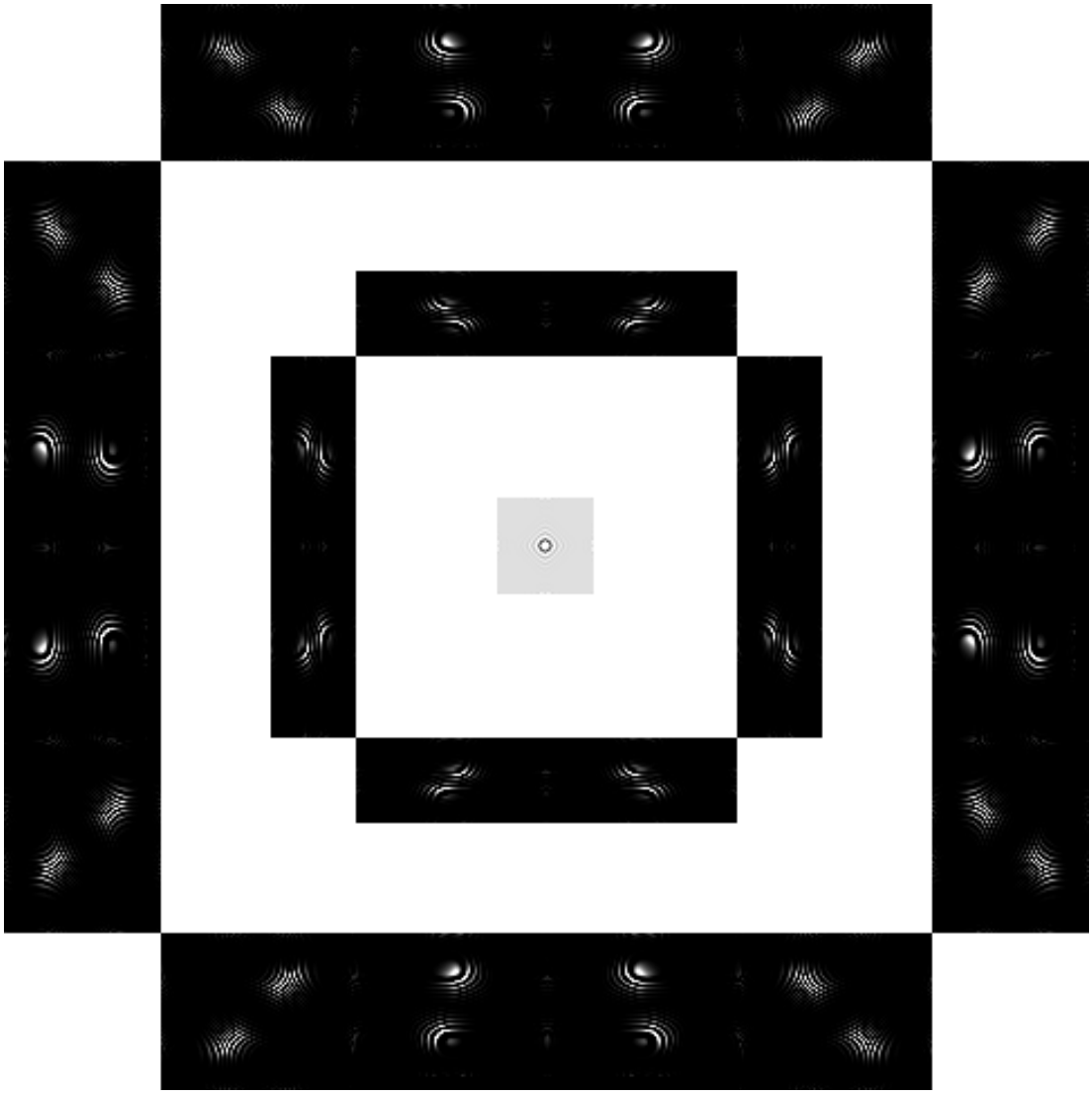


Figure 2.20: Steerable pyramid outputs of the zoneplate pattern, for 3 scales

2.6 Texton Filters

Textons are introduced by Julesz in 1981, as "the putative units of pre-attentive human texture perception" [49]. He described them as simple binary line segment stimulioriented segments, crossings and terminators, but did not provide an operational definition for gray-level images. Subsequently, texton theory fell into disfavor as a model of human texture discrimination as accounts based on spatial filtering with orientation and scale-selective mechanisms that could be applied to arbitrary gray-level images became popular. The term is reinvented in 1999 by Malik *et. al.* as clustered responses of an image to a filter bank [50]. They argue that the response to oriented odd-symmetric filter is not discriminative enough that visual cues such as edges, bars, or corners cannot be associated with the output of a single filter. Rather it is the characteristic of the outputs over scales, orientations and order of the filter that gives the relevant information. Following this logic, they consider outputs of these filters as points in a high dimensional space and try to find common characteristics for each class, via clustering. They define the centroid of resulting clusters, as textons. It turns out that textons do tend to correspond to visual cues such as oriented bars, terminators, etc. in an image. They propose that one can construct a texton dictionary by processing a large number of natural images, or we could find them adaptively in windows of images. In each case the K-means technique can be used. By mapping each pixel to the texton nearest to its vector of filter response

There are a number of filter banks constructed for texton formation that are multi-scale and direction-sensitive. Two of these, which are used in this thesis are small Leung-Malik (LMS) filter bank and Maximum Response 8 (MR8) filter bank. LM Filters are multi scale, multi orientation filter bank with 48 filters. It consists of first and second derivatives of Gaussians at 6 orientations and 3 scales making a total of 36; 8 Laplacian of Gaussian (LOG) filters; and 4 Gaussians. In LMS, the filters occur at basic scales $\sigma = \{1, \sqrt{2}, 2, 2\sqrt{2}\}$. The first and second derivative filters occur at the first three scales with an elongation factor of 3 (i.e. $\sigma_x = \sigma$ and $\sigma_x = 3\sigma$). The Gaussians occur at the four basic scales while the 8 LOG filters occur at σ and 3σ [51]. The LMS filter bank, and its response to

fireworks pattern are shown in Figures 2.21 and 2.22, respectively.

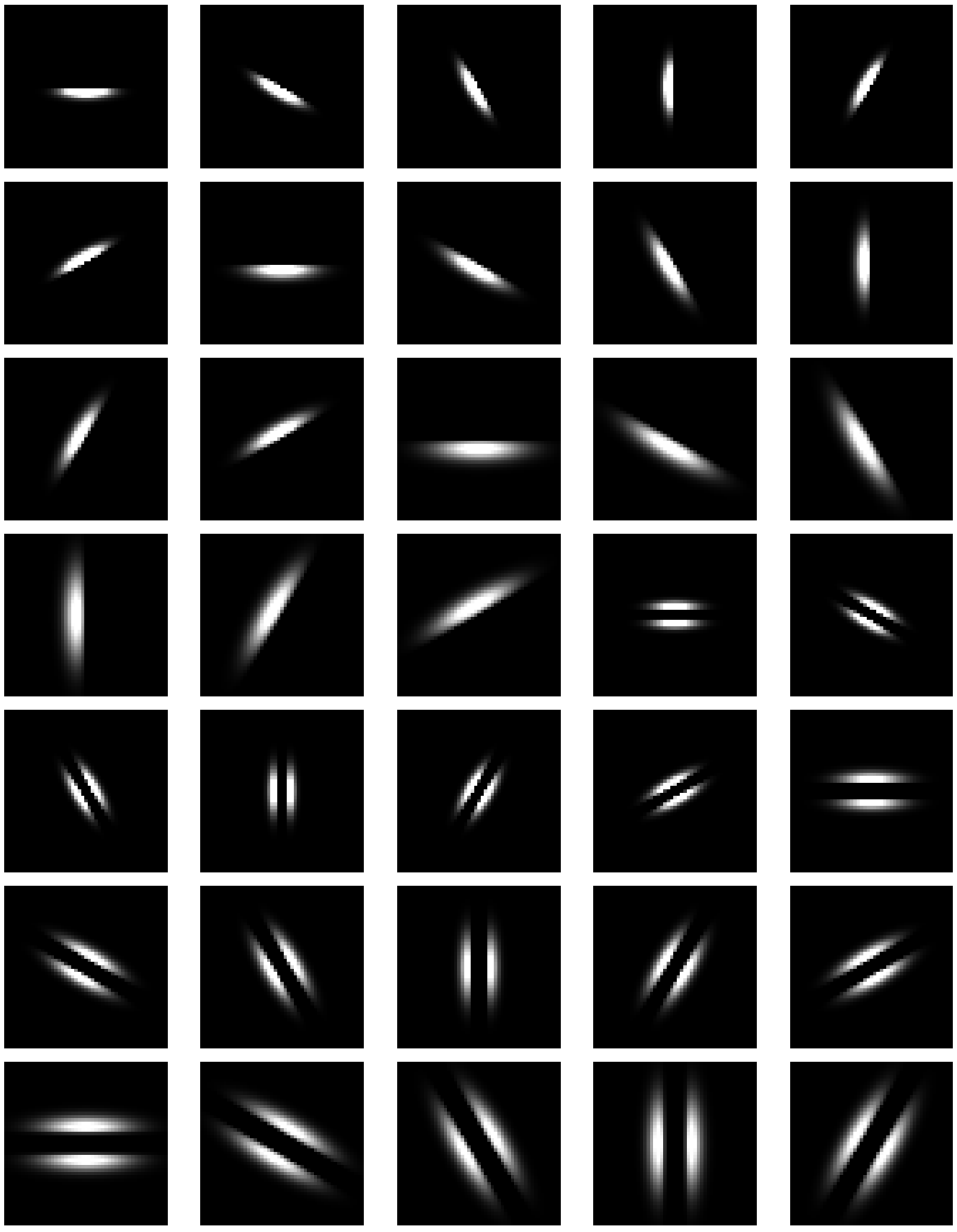
The root filter set (RFS) filter bank is shown in Figure 2.24 and consists of a Gaussian and a LoG both with $\sigma = 10$ pixels, an edge filter at 3 scales: $(\sigma_x, \sigma_y) = (1,3), (2,6), (4,12)$ and a bar filter at same scales [52]. The latter two filters are oriented, an as in LM occur at 6 orientations at each scale. In order to reduce number of dimensions, Maximum Response (MR8) filter bank is introduced consists of 38 filters but only 8 responses. The filter bank contains filters at multiple orientations but their outputs are 'collapsed' by recording only the maximum filter response across all orientations. This achieves rotation invariance. Measuring only the maximum response across orientations reduce the number of responses from 38 (6 orientations at 3 scales for 2 filters, plus 2 isotropic) to 8 (3 scales for 2 filters, plus 2 isotropic). An advantage of MR8 filter bank is that it can be applied using fast anisotropic Gaussian filtering [53], which results in significant decrease in computation time (see Table 2.2). The responses of RFS and MR8 filter banks to fireworks pattern can be seen in Figures 2.25 and 2.27, respectively. Although the MR8 filters is much faster than RFS filters, they display no directional selectivity.

2.7 Gabor Filters

2D Gabor filter are proposed by Daugman to model the response of direction sensitive simple cells in visual cortex [54]:

$$G(x,y) = \frac{1}{2\pi\sigma\beta} e^{-\pi[\frac{(x-x_0)^2}{\sigma^2} + \frac{(y-y_0)^2}{\beta^2}]} e^{i[\xi_0 x + \nu_0 y]} \tag{2.10}$$

where (x_0, y_0) is the center of receptive field in spatial domain, (ξ_0, ν_0) is the optimal spatial frequency of the filter in frequency domain, and (σ, β) are the standard deviations of the Gaussian function in x and y, respectively. Gabor function can be seen as a product of an elliptical Gaussian(first exponential in 2.10) and a complex planar wave(second exponential in 2.10).



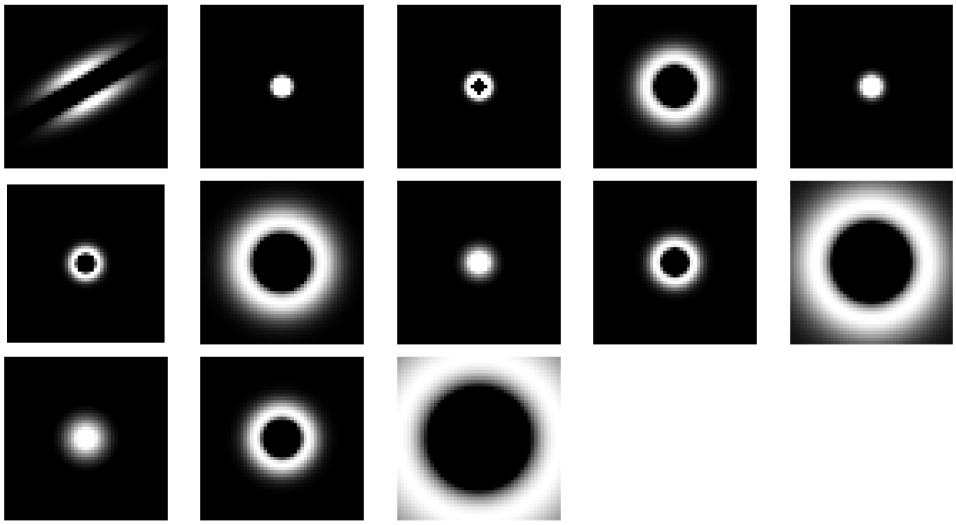


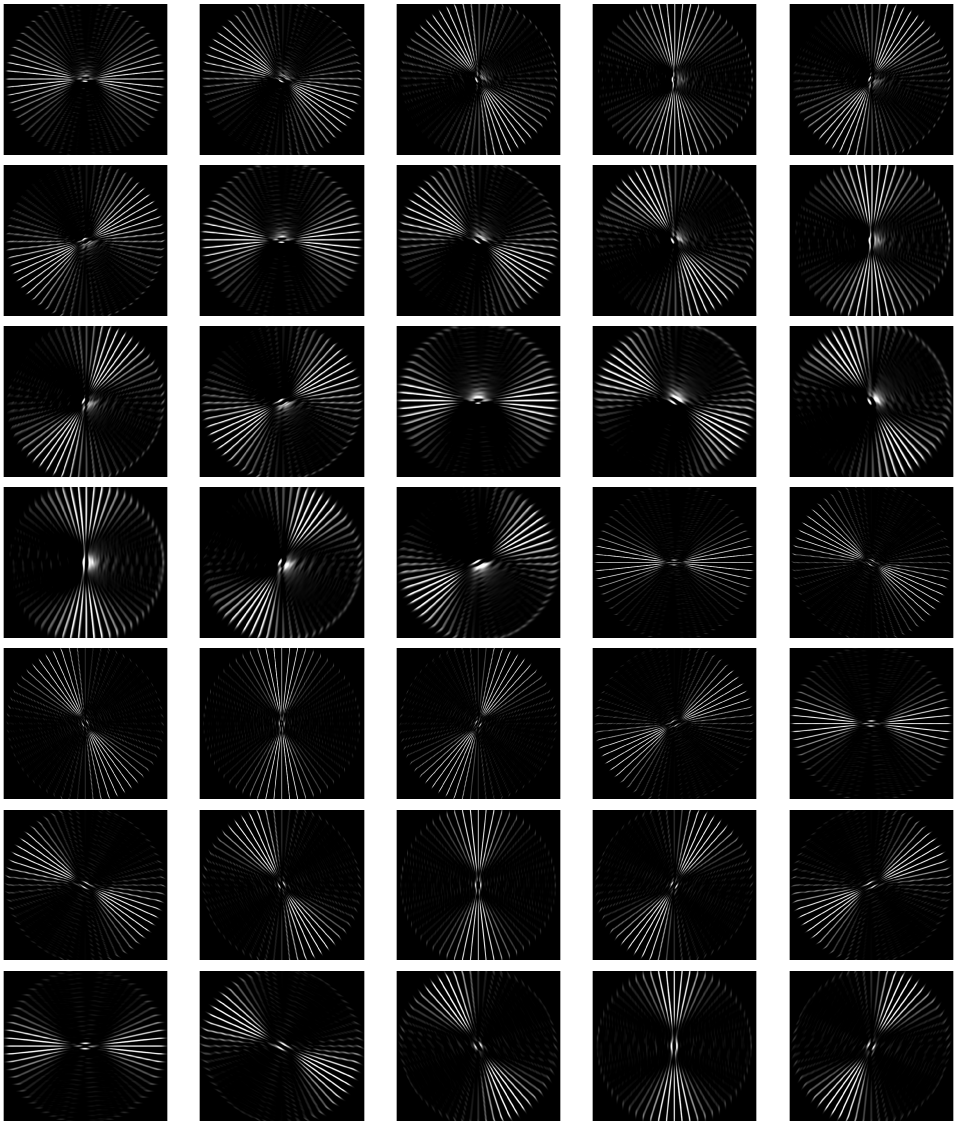
Figure 2.21: LM Filters

Instead of 2.10, slightly different parametrization is used in [55]:

$$g_{\lambda,\theta,\phi,\sigma,\gamma}(x,y) = \exp(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2})\cos(2\pi\frac{x'}{\lambda}+\phi) \tag{2.11}$$
$$x' = (x-\epsilon)\cos\theta + (y-\nu)\sin\theta$$
$$y' = (x-\epsilon)\sin\theta + (y-\nu)\cos\theta$$

where (ϵ, ν) is the center of filter in spatial domain, σ is the standard deviation of the Gaussian, γ is the spatial aspect ratio of the filter a, ϕ is the phase offset, θ is the orientation of the filter and λ is the wavelength of plane wave. Note that 2.11 corresponds to the real part of the reparametrized 2D complex Gabor function: $g_{\lambda,\theta,\phi,\sigma,\gamma}(x,y) = \exp(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2})\exp(i[2\pi\frac{x'}{\lambda}+\phi])$

Parametrization of 2.11 will be used throughout this thesis. Since Gabor filters can be complex, we must choose which filter output to use: convolve the image with real part, imaginary part, or convolve it with both filters and fuse two outputs using absolute value. Figure 2.29 shows the highest % 2 values of the three outputs, $\lambda = 5.4$, $\theta = \pi/8$. Output of complex filter is clearly more concentrated along the desired direction, whereas outputs of real and imaginary filters spread to adjoining angles. Therefore, although it doubles the computation time (due to doubling the number of convolutions), we choose to use the output



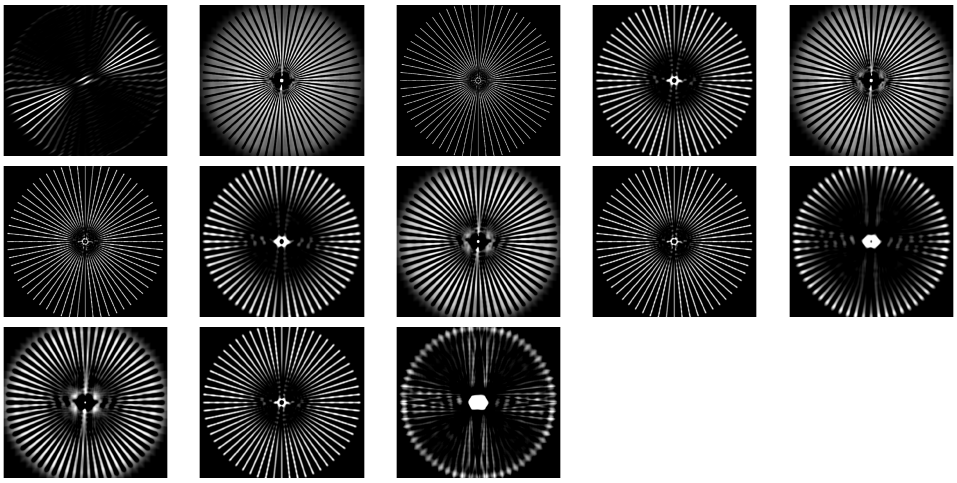


Figure 2.22: LM Responses to fireworks pattern

of the complex filter. Resulting filter output is :

$$\mathbf{Y} = abs(\mathbf{I} * \text{real}(g_{\lambda,\theta,\phi,\sigma,\gamma}) + i[\mathbf{I} * \text{imag}(g_{\lambda,\theta,\phi,\sigma,\gamma})]) \quad (2.12)$$

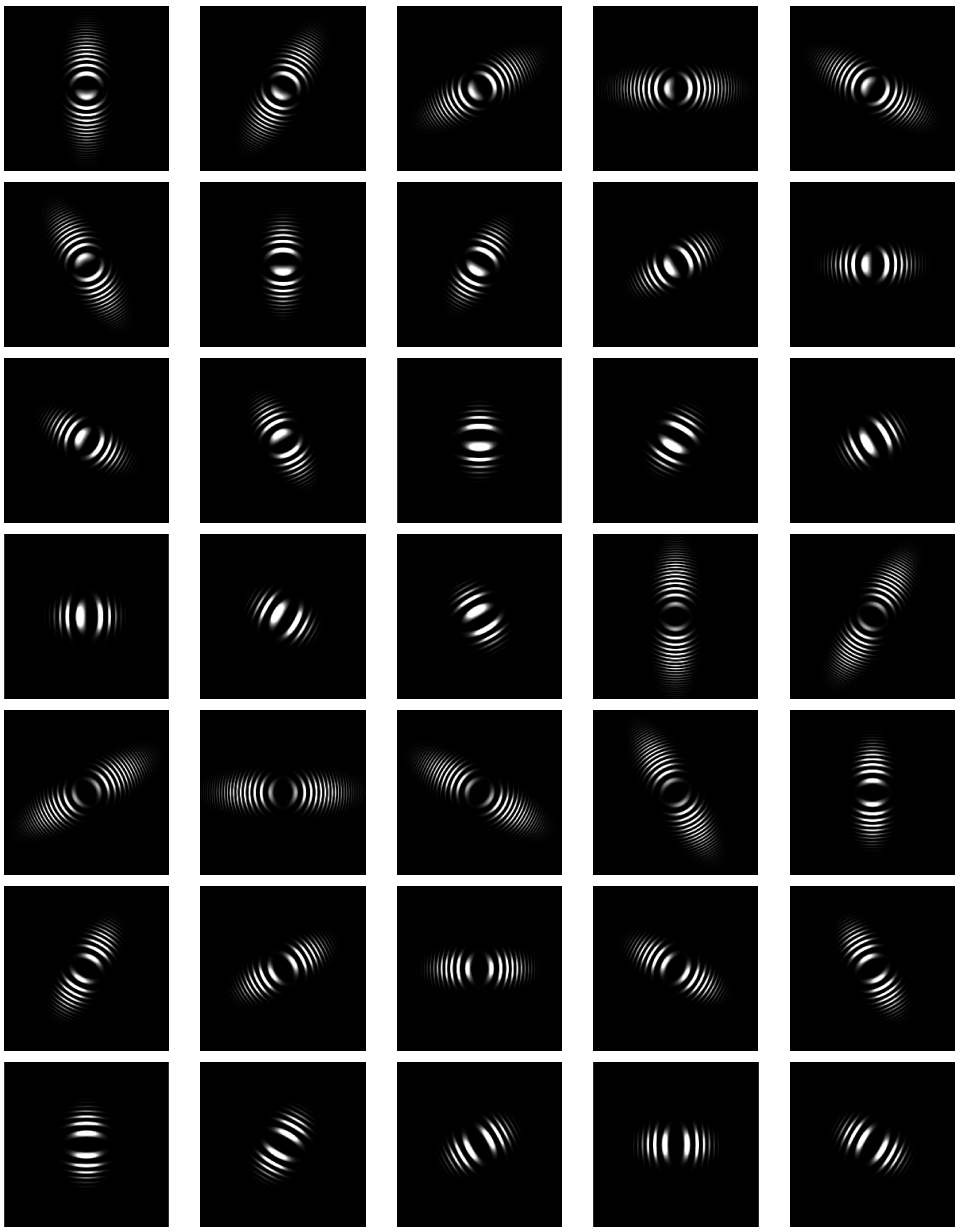
$$g_{\lambda,\theta,\phi,\sigma,\gamma}(x,y) = \exp(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}) \exp(i[2\pi\frac{x'}{\lambda} + \phi]) \quad (2.13)$$

$$x' = (x-\epsilon)\cos\theta + (y-\nu)\sin\theta$$

$$y' = (x-\epsilon)\sin\theta + (y-\nu)\cos\theta$$

Figures 2.30, 2.31 and 2.32 show the Gabor filters used in this thesis, and their responses to the fireworks and zoeplate pattern, respectively.

It has been found that spatial aspect ratio is limited in the range $0.23 < \gamma < 0.92$ [56]. In this thesis, we choose $\gamma = 0.5$. For rest of the parameters, we choose parameters as in [31]: $\sigma = 0.56$, $\lambda = [2.7, 4.1, 5.4]$ and $\theta = [\pi/8, 2\pi/8, \ldots, \pi]$. Values in Table 2.2 are measured with these parameters.



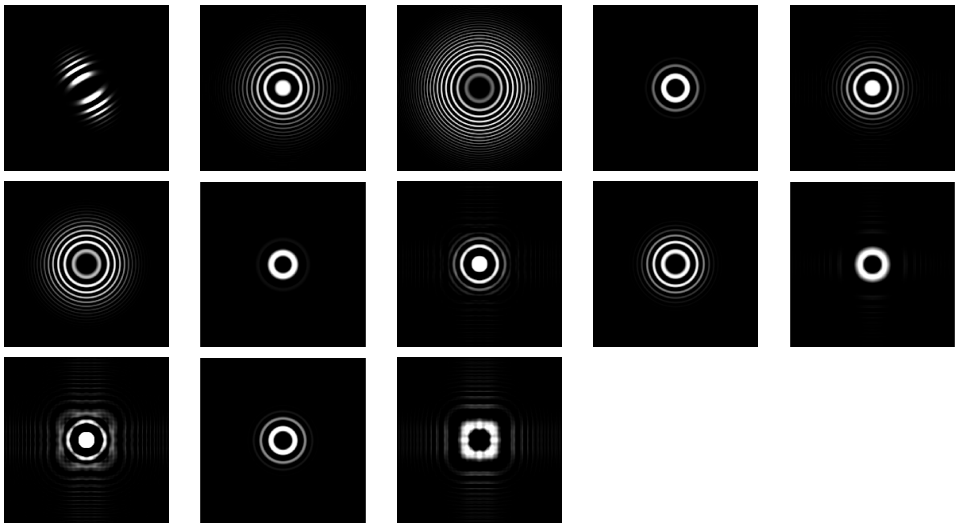


Figure 2.23: LM Responses to zoneplate pattern

2.8 Gray Level Co-Occurrence Matrices

Gray level co-occurrence matrices are first proposed as 'gray-tone spatial dependence matrices' by Haralick *et. al.*[12], to extract the spatial relationship which ne gray tones in image **I** have with one another. Haralick proposes that this texture-content information can be specified by matrix of relative frequencies P_{ij} with two neighbouring cells separated by distance d occur on the image with one in gray tone i and other in gray tone j. Assume for a rectangular gray level image with N_x horizontal pixels and N_y vertical pixels, so that the whole spatial domain can be spanned by product of $L_x = 1, \ldots N_x$ and $L_y = 1, \ldots N_y$. Moreover, assume that the image is quantized to N_θ levels. Then P_{ij} for angles can be formally defined as:

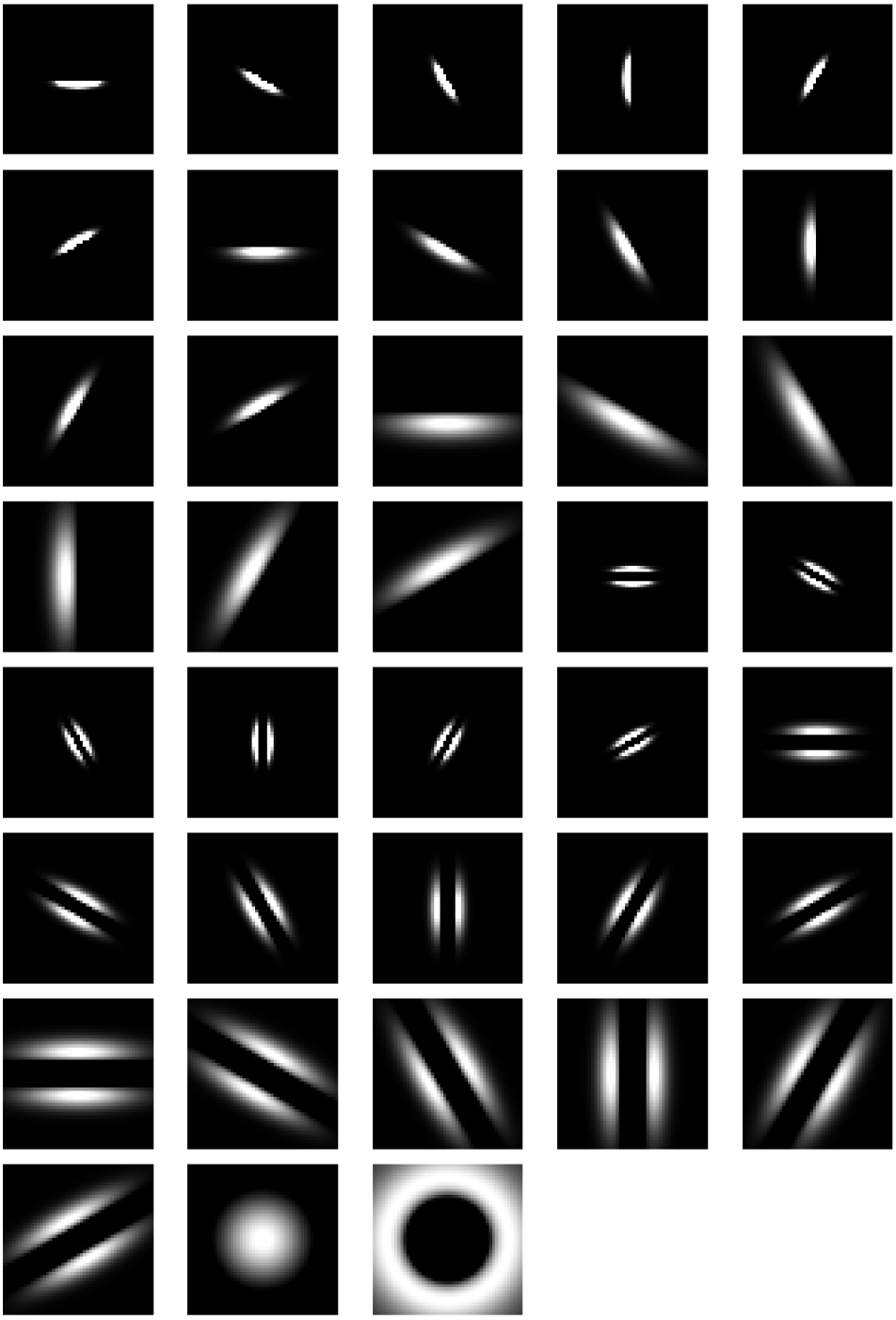


Figure 2.24: RFS Filters

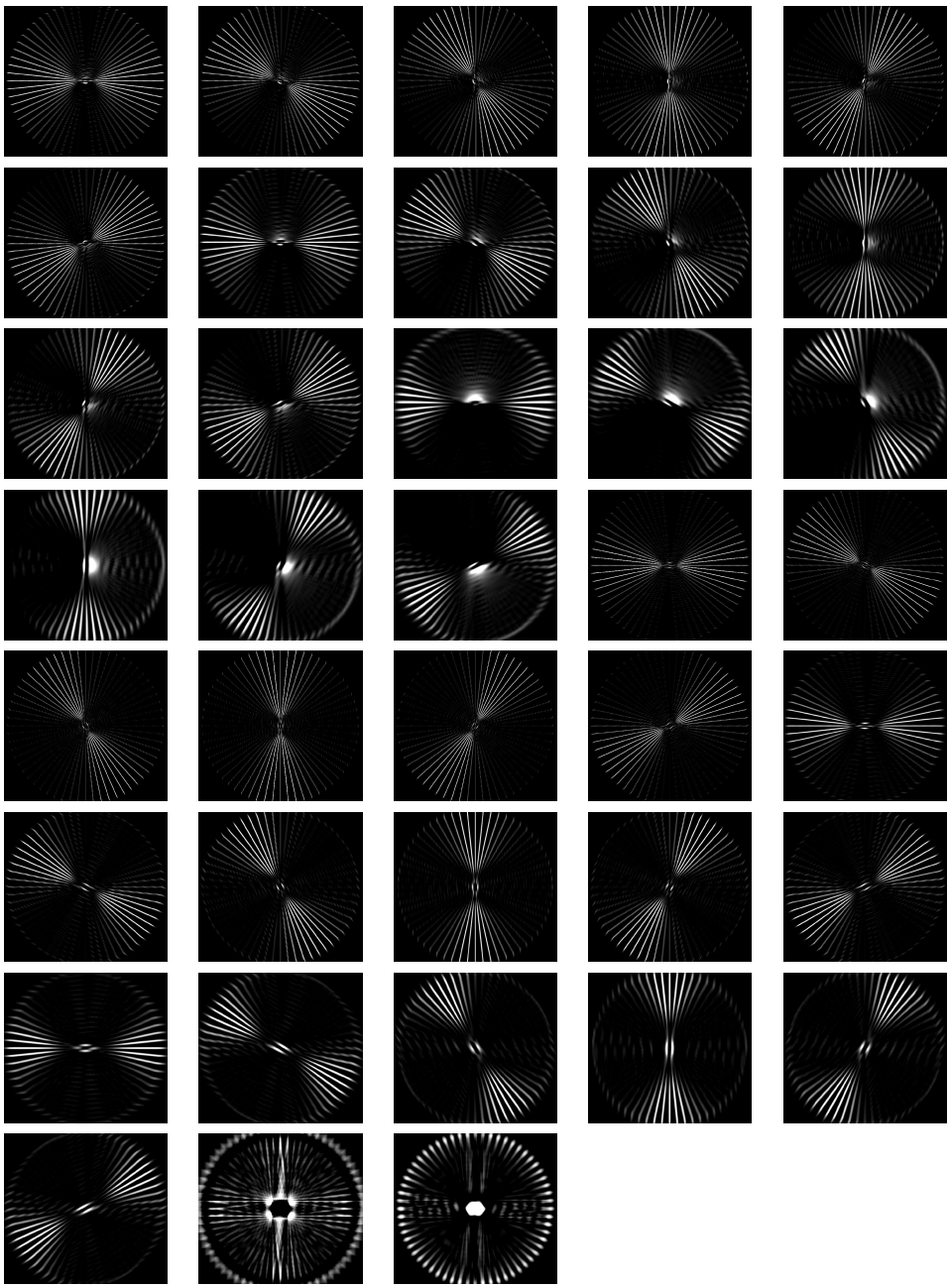


Figure 2.25: RFS Responses to fireworks pattern

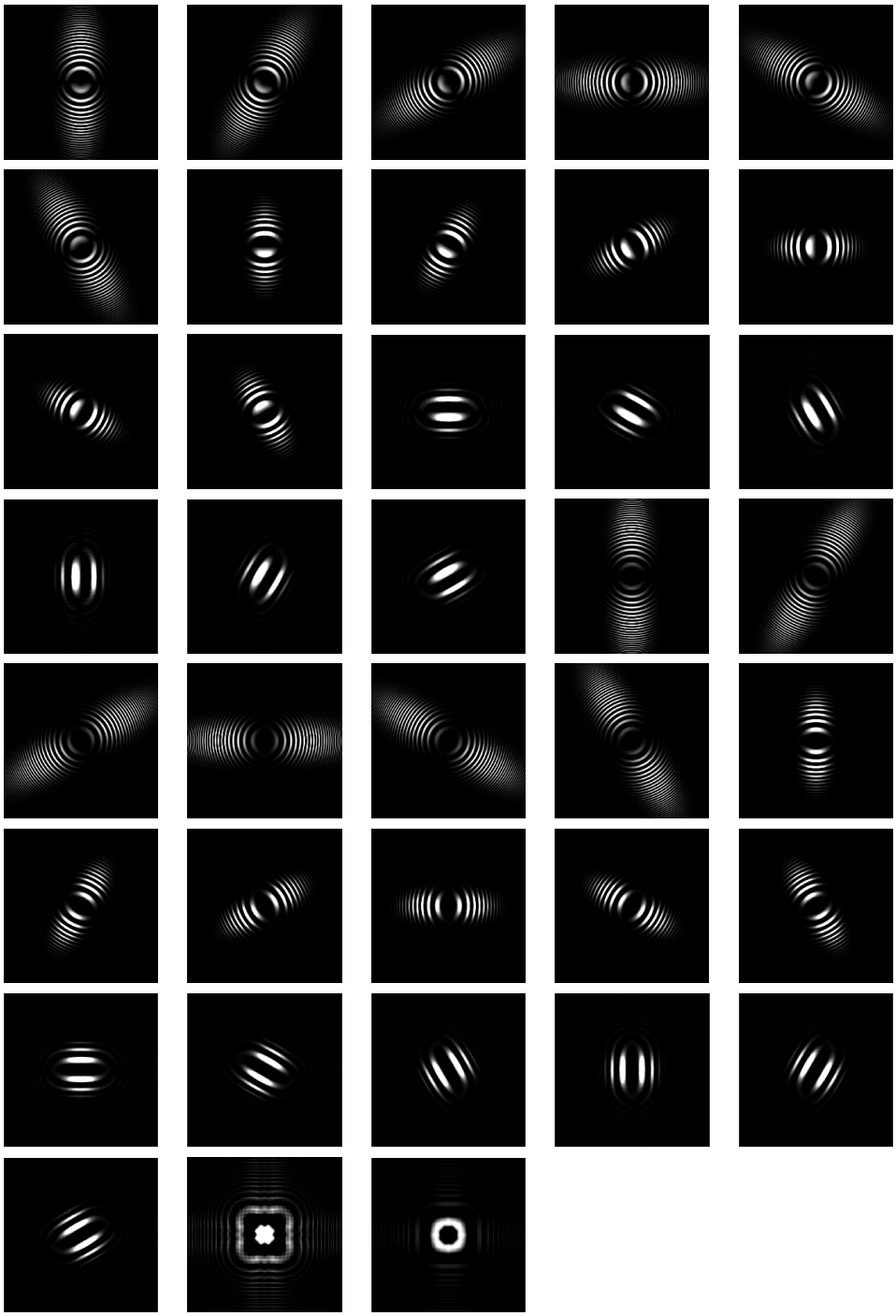


Figure 2.26: RFS Responses of zoneplate pattern

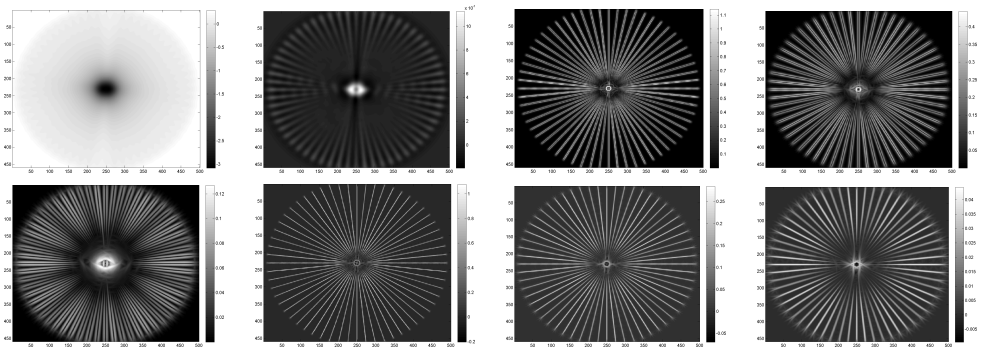


Figure 2.27: MR8 Responses to fireworks pattern

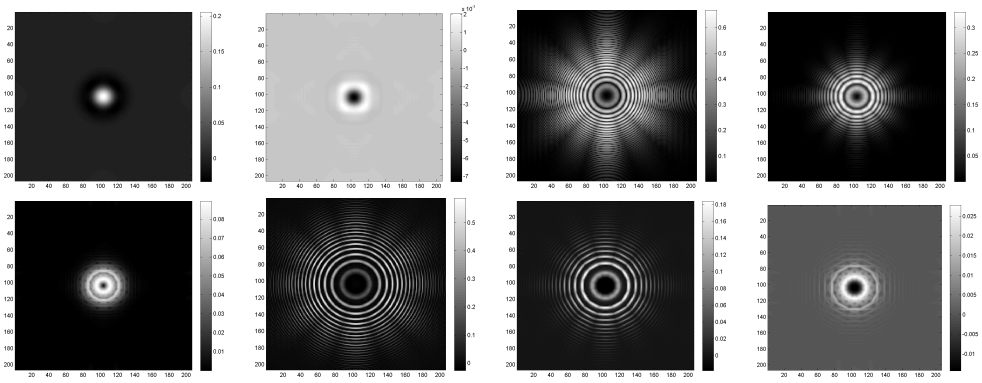
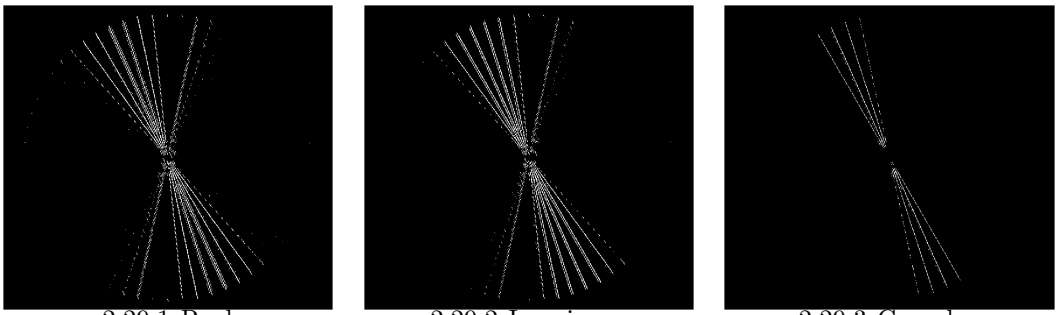
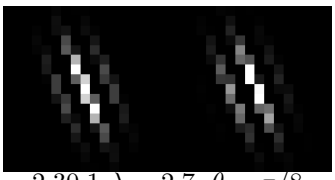


Figure 2.28: MR8 Responses to zoneplate pattern

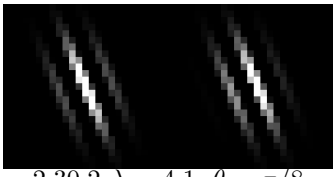


2.29.1 Real 2.29.2 Imaginary 2.29.3 Complex

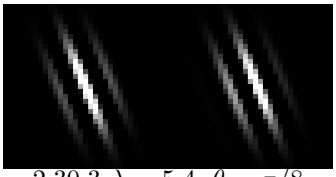
Figure 2.29: Highest % 2 Responses of real, imaginary and complex Gabor Filters to fireworks pattern, for $\lambda = 5.4$, $\theta = \pi/8$



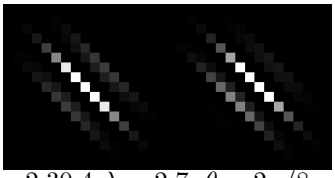
2.30.1 $\lambda = 2.7$, $\theta = \pi/8$



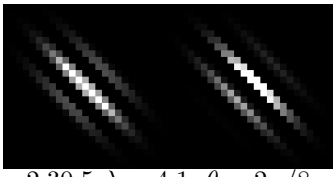
2.30.2 $\lambda = 4.1$, $\theta = \pi/8$



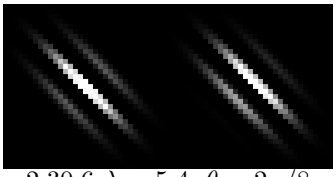
2.30.3 $\lambda = 5.4$, $\theta = \pi/8$



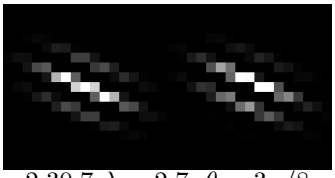
2.30.4 $\lambda = 2.7$, $\theta = 2\pi/8$



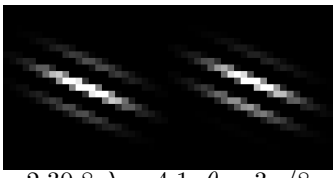
2.30.5 $\lambda = 4.1$, $\theta = 2\pi/8$



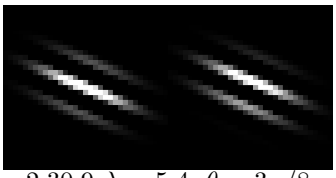
2.30.6 $\lambda = 5.4$, $\theta = 2\pi/8$



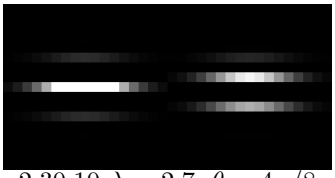
2.30.7 $\lambda = 2.7$, $\theta = 3\pi/8$



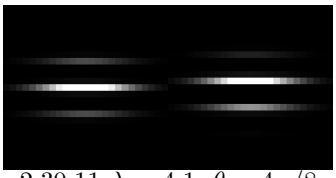
2.30.8 $\lambda = 4.1$, $\theta = 3\pi/8$



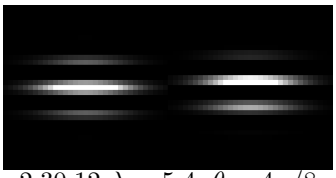
2.30.9 $\lambda = 5.4$, $\theta = 3\pi/8$



2.30.10 $\lambda = 2.7$, $\theta = 4\pi/8$



2.30.11 $\lambda = 4.1$, $\theta = 4\pi/8$



2.30.12 $\lambda = 5.4$, $\theta = 4\pi/8$

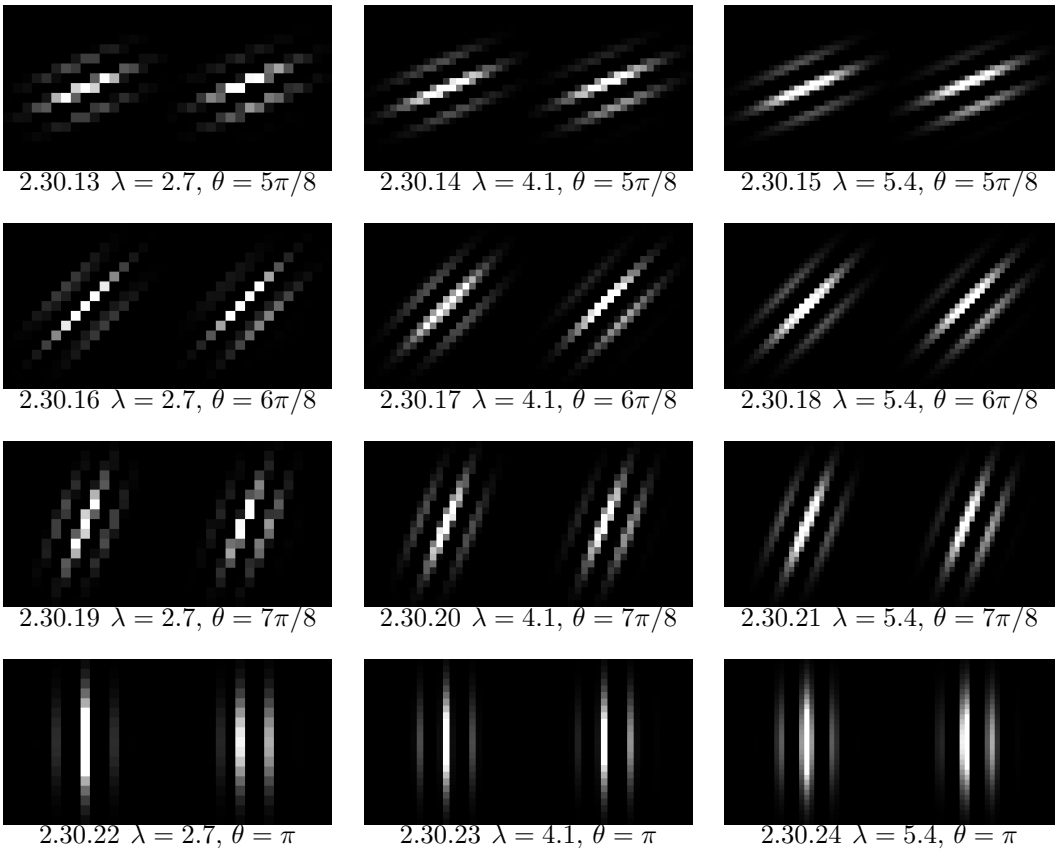
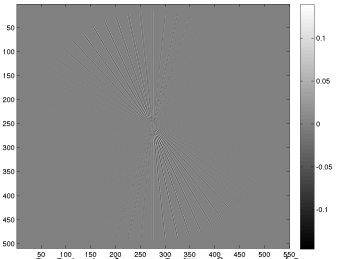
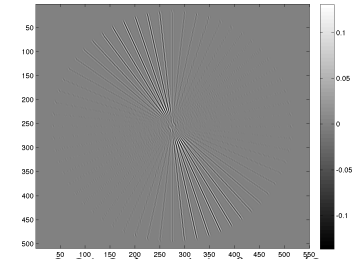


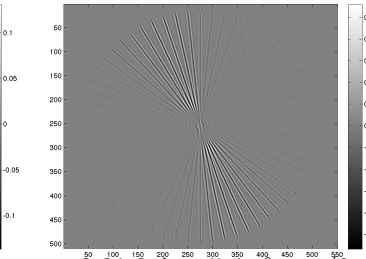
Figure 2.30: Real(left filter in each image) and imaginary(right filter in each image) Gabor Filters, for $\lambda = [2.7, 4.1, 5.4]$ and $\theta = [\pi/8, 2\pi/8, \ldots, \pi]$.



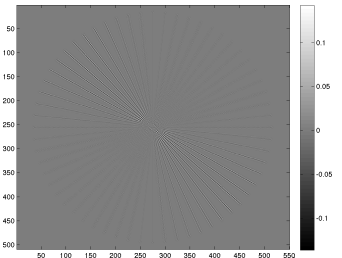
2.31.1 $\lambda = 2.7$, $\theta = \pi/8$



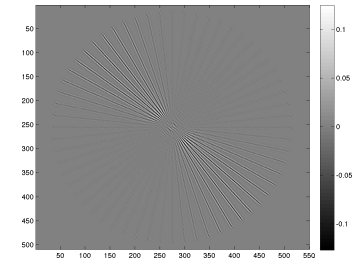
2.31.2 $\lambda = 4.1$, $\theta = \pi/8$



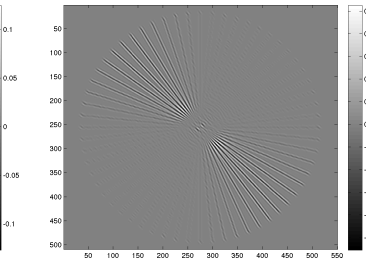
2.31.3 $\lambda = 5.4$, $\theta = \pi/8$



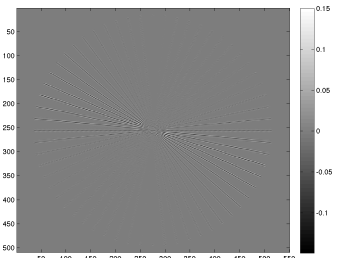
2.31.4 $\lambda = 2.7$, $\theta = 2\pi/8$



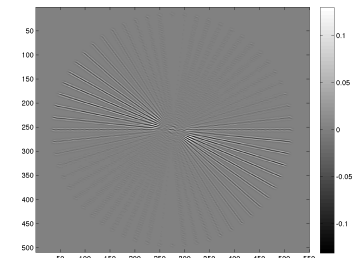
2.31.5 $\lambda = 4.1$, $\theta = 2\pi/8$



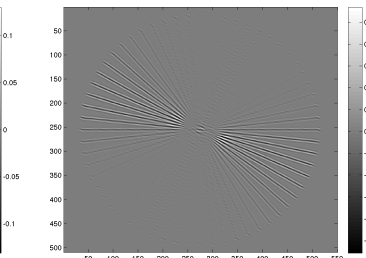
2.31.6 $\lambda = 5.4$, $\theta = 2\pi/8$



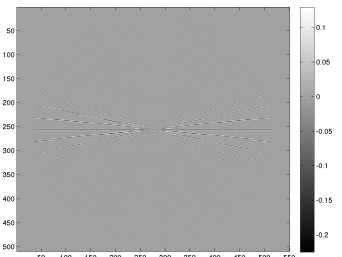
2.31.7 $\lambda = 2.7$, $\theta = 3\pi/8$



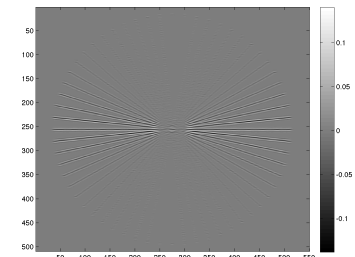
2.31.8 $\lambda = 4.1$, $\theta = 3\pi/8$



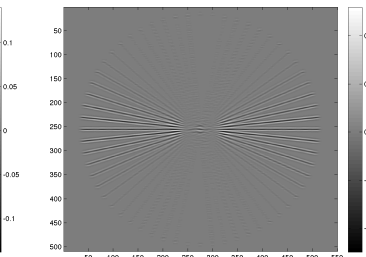
2.31.9 $\lambda = 5.4$, $\theta = 3\pi/8$



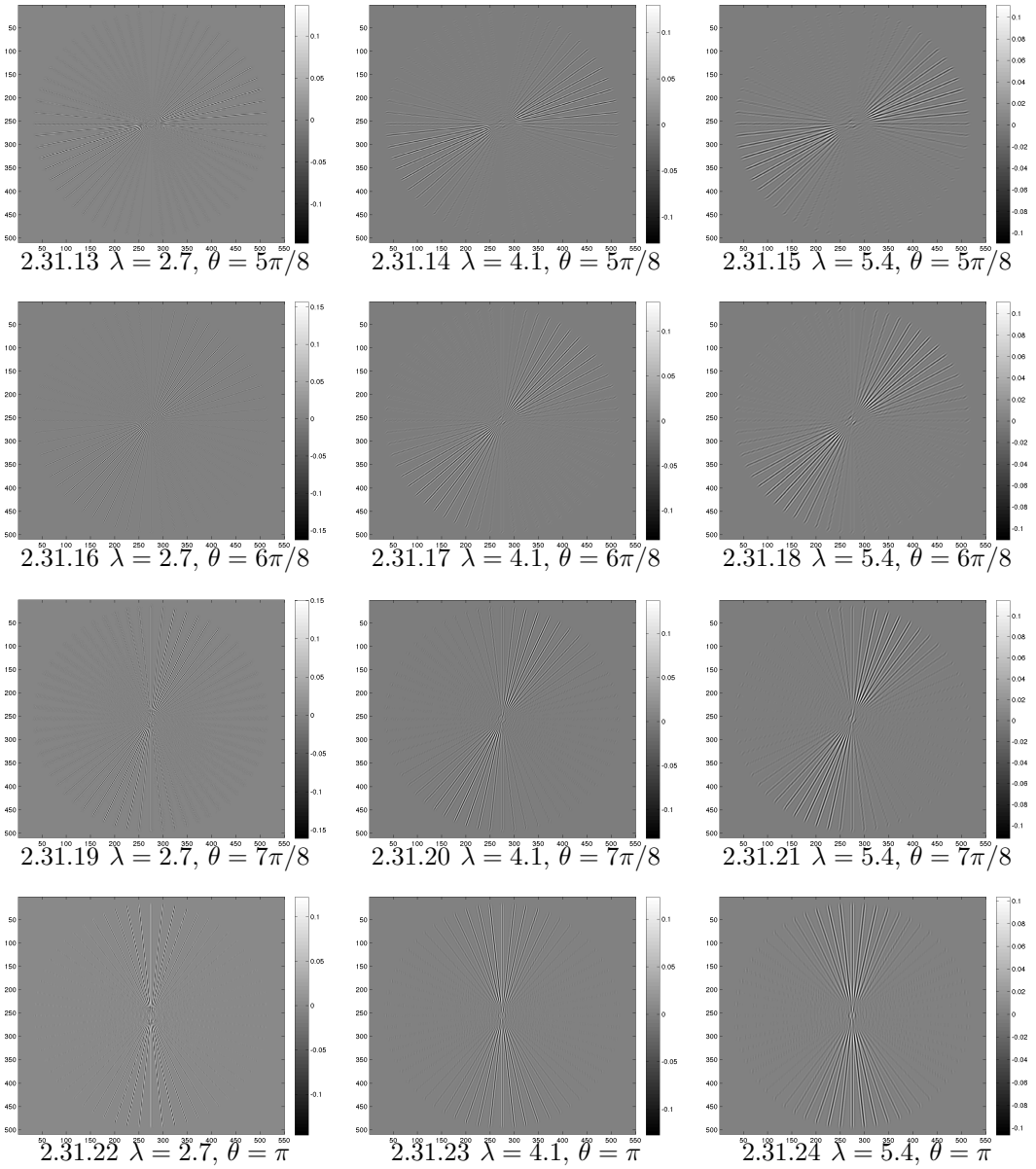
2.31.10 $\lambda = 2.7$, $\theta = 4\pi/8$



2.31.11 $\lambda = 4.1$, $\theta = 4\pi/8$



2.31.12 $\lambda = 5.4$, $\theta = 4\pi/8$



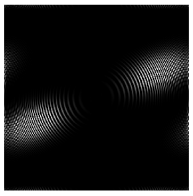
2.31.13 $\lambda = 2.7$, $\theta = 5\pi/8$ 2.31.14 $\lambda = 4.1$, $\theta = 5\pi/8$ 2.31.15 $\lambda = 5.4$, $\theta = 5\pi/8$

2.31.16 $\lambda = 2.7$, $\theta = 6\pi/8$ 2.31.17 $\lambda = 4.1$, $\theta = 6\pi/8$ 2.31.18 $\lambda = 5.4$, $\theta = 6\pi/8$

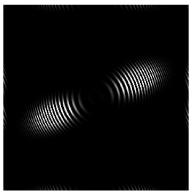
2.31.19 $\lambda = 2.7$, $\theta = 7\pi/8$ 2.31.20 $\lambda = 4.1$, $\theta = 7\pi/8$ 2.31.21 $\lambda = 5.4$, $\theta = 7\pi/8$

2.31.22 $\lambda = 2.7$, $\theta = \pi$ 2.31.23 $\lambda = 4.1$, $\theta = \pi$ 2.31.24 $\lambda = 5.4$, $\theta = \pi$

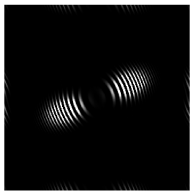
Figure 2.31: Gabor Filter responses to fireworks pattern, for $\lambda = [2.7, 4.1, 5.4]$ and $\theta = [\pi/8, 2\pi/8, \ldots, \pi]$



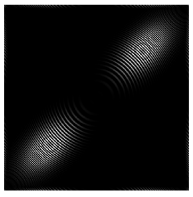
2.32.1 $\lambda = 2.7$, $\theta = \pi/8$



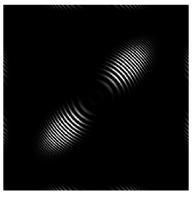
2.32.2 $\lambda = 4.1$, $\theta = \pi/8$



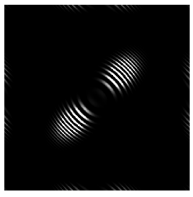
2.32.3 $\lambda = 5.4$, $\theta = \pi/8$



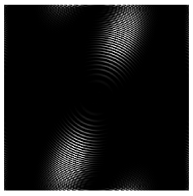
2.32.4 $\lambda = 2.7$, $\theta = 2\pi/8$



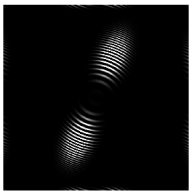
2.32.5 $\lambda = 4.1$, $\theta = 2\pi/8$



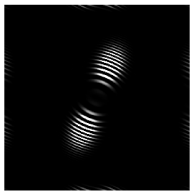
2.32.6 $\lambda = 5.4$, $\theta = 2\pi/8$



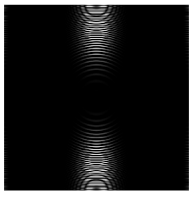
2.32.7 $\lambda = 2.7$, $\theta = 3\pi/8$



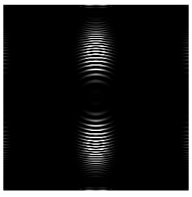
2.32.8 $\lambda = 4.1$, $\theta = 3\pi/8$



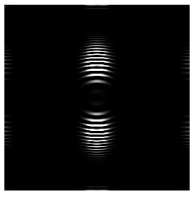
2.32.9 $\lambda = 5.4$, $\theta = 3\pi/8$



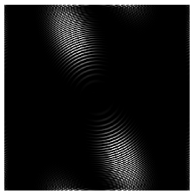
2.32.10 $\lambda = 2.7$, $\theta = 4\pi/8$



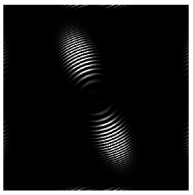
2.32.11 $\lambda = 4.1$, $\theta = 4\pi/8$



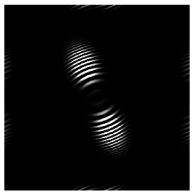
2.32.12 $\lambda = 5.4$, $\theta = 4\pi/8$



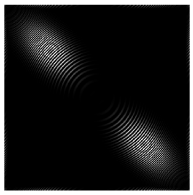
2.32.13 $\lambda = 2.7$, $\theta = 5\pi/8$



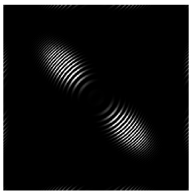
2.32.14 $\lambda = 4.1$, $\theta = 5\pi/8$



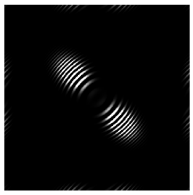
2.32.15 $\lambda = 5.4$, $\theta = 5\pi/8$



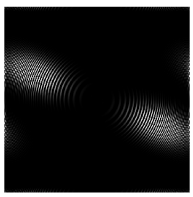
2.32.16 $\lambda = 2.7$, $\theta = 6\pi/8$



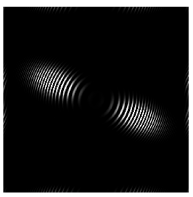
2.32.17 $\lambda = 4.1$, $\theta = 6\pi/8$



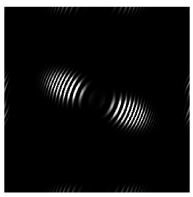
2.32.18 $\lambda = 5.4$, $\theta = 6\pi/8$



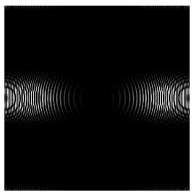
2.32.19 $\lambda = 2.7$, $\theta = 7\pi/8$



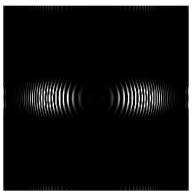
2.32.20 $\lambda = 4.1$, $\theta = 7\pi/8$



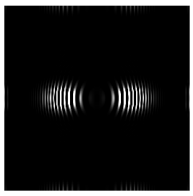
2.32.21 $\lambda = 5.4$, $\theta = 7\pi/8$



2.32.22 $\lambda = 2.7$, $\theta = \pi$



2.32.23 $\lambda = 4.1$, $\theta = \pi$



2.32.24 $\lambda = 5.4$, $\theta = \pi$

Figure 2.32: Gabor Filter responses to zoneplate pattern, for $\lambda = [2.7, 4.1, 5.4]$ and $\theta = [\pi/8, 2\pi/8, \ldots, \pi]$

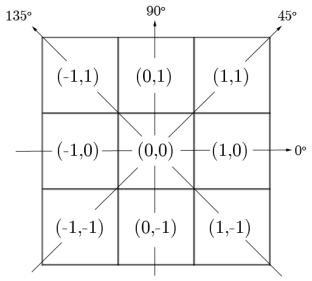


Figure 2.33: Immediate neighbours(d=1) of a pixel, corresponding offset values and corresponding angles

$$P(i,j,d,0^\circ) = \#\{((k,l),(m,n)) \in (L_y \times L_x) \times (L_y \times L_x) \quad (2.14)$$
$$|k-m=0, |l-n| = d, I(k,l) = i, I(m,n) = j\}$$
$$P(i,j,d,45^\circ) = \#\{((k,l),(m,n)) \in (L_y \times L_x) \times (L_y \times L_x) \quad (2.15)$$
$$|(k-m=d, l-n=-d) \vee (k-m=-d, l-n=d), I(k,l) = i, I(m,n) = j\}$$
$$P(i,j,d,90^\circ) = \#\{((k,l),(m,n)) \in (L_y \times L_x) \times (L_y \times L_x) \quad (2.16)$$
$$||k-m| = 0, l-n=0, I(k,l) = i, I(m,n) = j\}$$
$$P(i,j,d,135^\circ) = \#\{((k,l),(m,n)) \in (L_y \times L_x) \times (L_y \times L_x) \quad (2.17)$$
$$|(k-m=d, l-n=d) \vee (k-m=-d, l-n=-d), I(k,l) = i, I(m,n) = j\}$$

Once P_{ij} is formed, there are number of features (14, to be exact) that can be calculated from it. Some of such textures are

$$f_1 = \sum_{i=1}^{N_\theta} \sum_{j=1}^{N_\theta} \left(\frac{P(i,j)}{R} \right)^2 \quad (2.18)$$

$$f_2 = \sum_{n=0}^{N_\theta - 1} n^2 \left\{ \sum_{|i-j|=n} \left(\frac{P(i,j)}{R} \right) \right\} \quad (2.19)$$

$$f_3 = \frac{f1 - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (2.20)$$

where $\mu_x, \mu_y, \sigma_x, \sigma_y$ are the means an standard deviations of the marginal

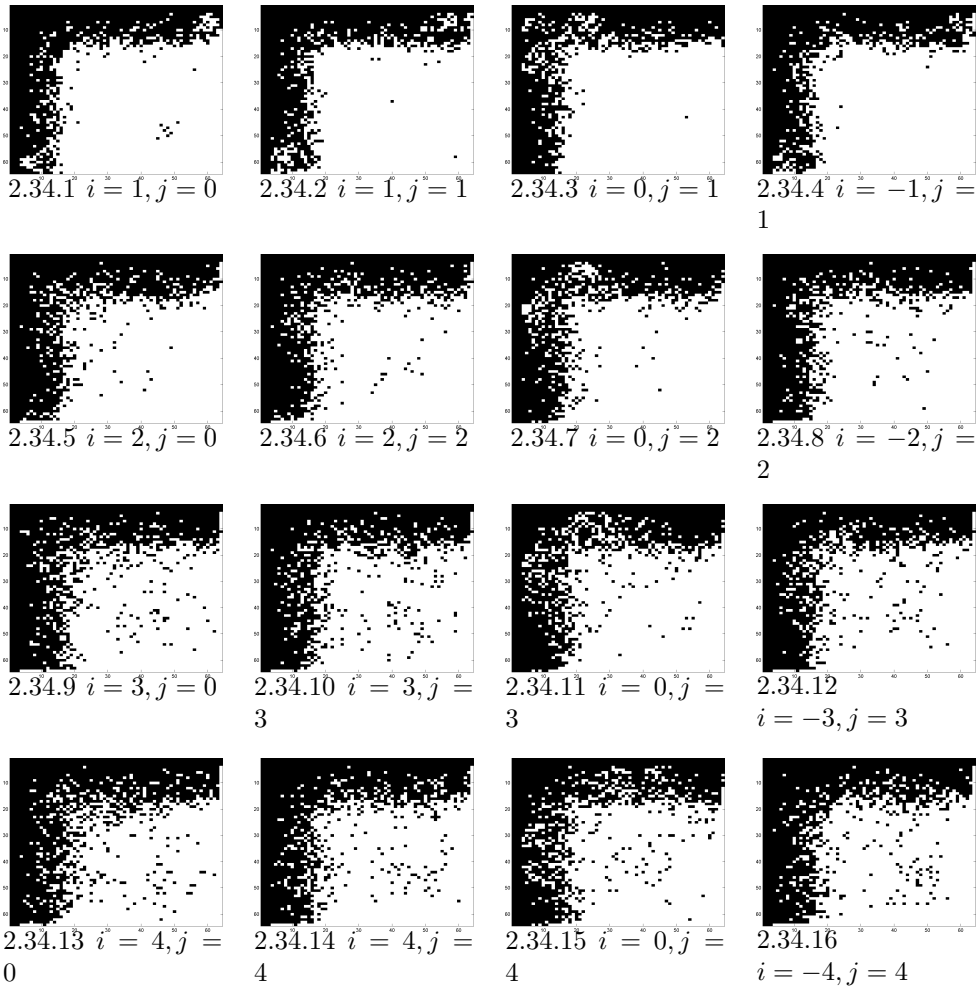
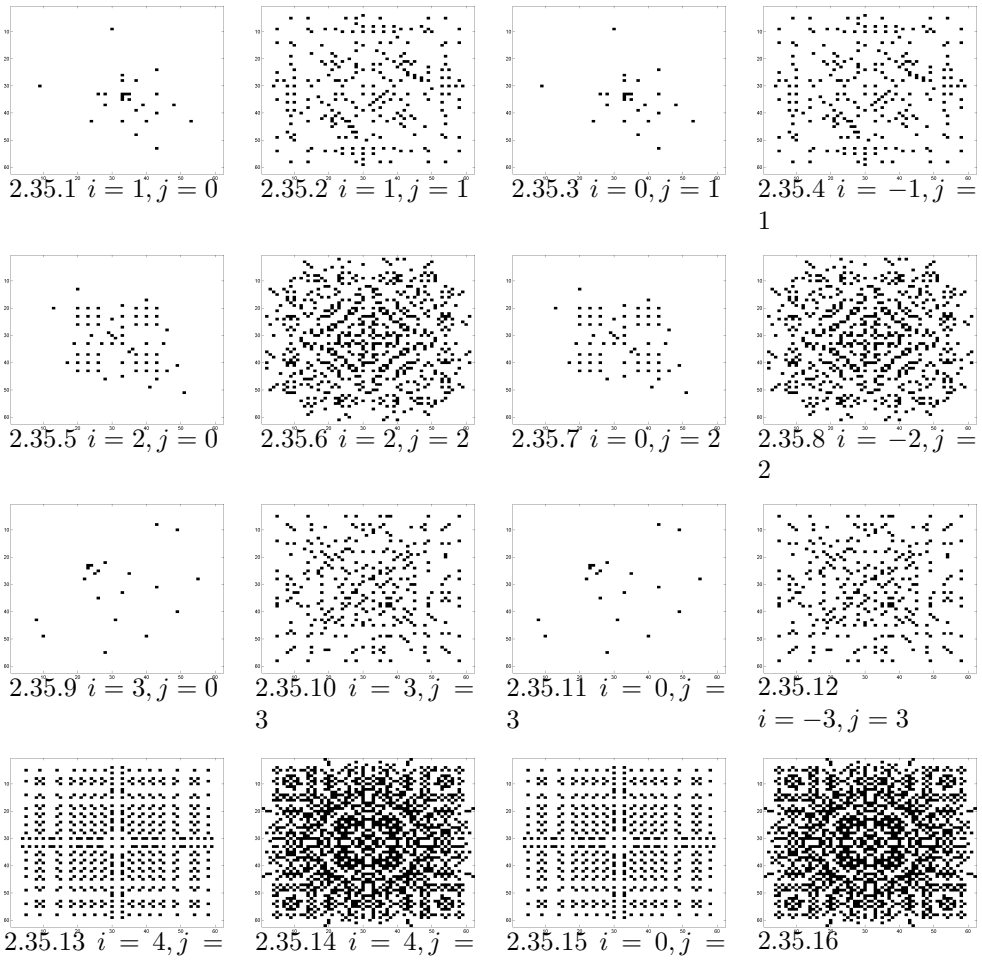


Figure 2.34: Non-zero elements of GLCM of fireworks pattern, for $d = 1, \ldots, 4$ and $\theta = \{0°, 45°, 90°, 135°\}$

distributions associated with $P(i, j)/R$, and R is a normalizing constant. f_1 is known as angular second moment, f_2 is contrast and f_3 is correlation. As stated, there are number of other features such as variance, inverse difference moment, entropy, etc. In this thesis, we calculate P(i,j) for four angles in Figure 2.33 using $d = \{1, 2, 3, 4\}$ and $N_\theta = 64$. For feature we use the L_0 norm of $P(i, j)$, i.e the number of non-zero elements in $P(i, j)$. The non-zero elements of GLCM for each d and angle are presented in Figures 2.34 and 2.35



2.35.1 $i = 1, j = 0$ 2.35.2 $i = 1, j = 1$ 2.35.3 $i = 0, j = 1$ 2.35.4 $i = -1, j = 1$

2.35.5 $i = 2, j = 0$ 2.35.6 $i = 2, j = 2$ 2.35.7 $i = 0, j = 2$ 2.35.8 $i = -2, j = 2$

2.35.9 $i = 3, j = 0$ 2.35.10 $i = 3, j = 3$ 2.35.11 $i = 0, j = 3$ 2.35.12 $i = -3, j = 3$

2.35.13 $i = 4, j = 0$ 2.35.14 $i = 4, j = 4$ 2.35.15 $i = 0, j = 4$ 2.35.16 $i = -4, j = 4$

Figure 2.35: Non-zero elements of GLCM of zoneplate pattern, for $d = 1, \ldots, 4$ and $\theta = \{0°, 45°, 90°, 135°\}$

2.9 Comparison of Feature extraction speeds

These tests are done on a computer with Intel i7-4700MQ CPU and 16 GB memory. Values presented in Table 2.2 are average times over 10 runs.

Table 2.2: Time required for each feature to be extracted from a $N \times N$ image, for $N = [512, 1024, 2048]$

Feature	Required time per $N \times N$ sample (s)		
	N=512	N=1024	N=2048
CWT	0.0615	0.277	1.2129
Curvelet	0.1863	0.7188	3.3451
Contourlet	0.178	0.6051	2.5417
GLCM	0.3643	1.6249	6.6618
Gabor	0.9655	3.9268	16.1438
Pyramid	0.2714	1.4155	6.1437
Dir Fil(3)	0.0323	0.1343	0.5447
Dir. Fil(4)	0.0318	0.1339	0.5432
MR8	0.2083	0.8757	3.4984
LM	2.1083	7.2679	32.69
RFS	1.6273	5.8382	25.7325

Table 2.3: Each entry is divided by the smallest entry in the column, for $N = [512, 1024, 2048]$

Feature	Each entry is divided by the smallest entry in the column		
	N=512	N=1024	N=2048
CWT	1.93	2.07	2.23
Curvelet	5.86	5.37	6.16
Contourlet	5.60	4.52	4.68
GLCM	11.46	12.14	12.26
Gabor	30.36	29.33	29.72
Pyramid	8.53	10.57	11.31
Dir Fil(3)	1.02	1.00	1.00
Dir. Fil(4)	1.00	1.00	1.00
MR8	6.55	6.54	6.44
LM	66.30	54.28	60.18
RFS	51.17	43.60	47.37

Chapter 3

Experiments on Font Recognition Datasets

In this section we provide a detailed experimental evaluation conducted to test the effectiveness of multiresolution directional feature extraction algorithms on font recognition, and comparisons with related studies in the literature. In all experiments, the accuracy of the model for recognizing different fonts are computed over 10-fold cross validation. First we describe the feature extraction framework in which the feature extraction algorithms are utilized, then we present the results.

3.1 Feature extraction and classification framework for Font Recognition problem

In a proper document analysis application, empty areas and printed areas of each document should be separated, and only printed areas should be fed to the next stage of the application. Since this process is outside the scope of this thesis, we assume the printed areas are already extracted and the images in datasets only contain printed areas. It is possible that each printed area contains more

than one font (or more than one style of the same font), and it is desired that the recognition system should be able to discriminate between these differently printed words. Therefore more than a single decision per input is required. To implement this, we divide the input image into a grid, feed each window to feature extraction algorithms and give separate decisions for each window.

The nature of text images are binary: letters and be seen as foreground and empty spaces between characters, words or paragraphs as background. The images, however are usually RGB or grayscale. Therefore, converting images to binary before feature extraction aids the process. For conversion we use simple thresholding with threshold calculated with Otsu's method[57].

By definition, all feature extraction algorithms output directional features as a filter response for each scale-direction pair. These outputs cannot be used directly as features because they are variant to size of input window, scale, small perturbations in input image such as translation or rotation. In order to make the feature more standardized and more robust to these factors, we use mean and standard deviation of filter outputs for each scale-direction pair. If we use filter outputs of a 3-scale and 6-directional feature extraction algorithm directly, we cannot say anything about the size of our feature vector-it depends on the input image. Also, even if we assume the input window is just 24×24 pixels, the feature size would be $6 \times 24 \times 24 + 6 \times 12 \times 12 + 6 \times 6 \times 6 = 42552$, which is too large to classify efficiently. If we use mean and standard deviation of filter outputs for the same algorithm, we easily see that the feature size is $2 \times 3 \times 6 = 36$ regardless of the input. Then we classify the extracted features using Support Vector Machines (SVM) with Radial Basis Function (RBF) as kernel function. The accuracy of the system, as mentioned before, is measured by 10-fold cross validation, which is a very standard method for measuring the accuracy of classification in the literature.

Table 3.1: Recognition rates of a classifier trained with features extracted from 96x96 sized windows (using CWT features)

Font	Style	Subsample Size					
		96×96	144×144	192×192	240×240	288×288	336×336
Correct	Correct	93.14	95.98	98.78	99.48	100	100
Correct	Wrong	1.76	1.55	0.41	0.00	0.00	0.00
Wrong	Correct	3.52	1.99	0.65	0.52	0.00	0.00
Wrong	Wrong	1.57	0.49	0.16	0.00	0.00	0.00

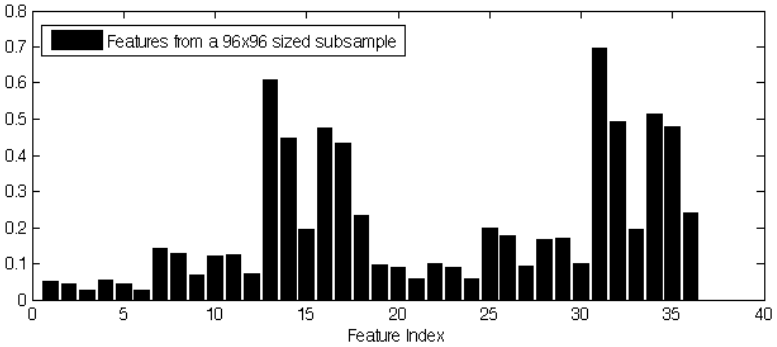
3.1.1 Choosing window size

Since the decisions are generated per window, it is desirable that the window size is chosen as small as possible. However, the height of the window should be at least larger than the height of the characters in the sample, because any smaller window will encapsulate only a part of the characters and cannot capture the true characteristics of the font.

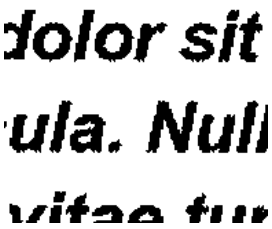
The statistical features we use (mean and standard deviation) allow us a degree of freedom once that lower bound is passed. These features capture a font-style pair's characteristics very similarly regardless of window size. Figure 3.1 shows the features extracted from 3 differently sized windows of Arial bold text. As it can be seen from Figure 3.1 , features of a 96×96 sized window are very similar to features of a 192×192 or 288×288 sized window. Therefore, a classifier trained with features extracted from 96×96 sized windows can easily classify features extracted from 192×192 or 288×288 sized windows.

To demonstrate this, we did several tests on Dataset 2. A classifier is trained with features extracted from 96×96 sized windows, and then used to classify features extracted from 96×96, 144×144, 192×192, 240×240, 288×288 and 336×336 sized windows. The results are presented in Table 3.1.

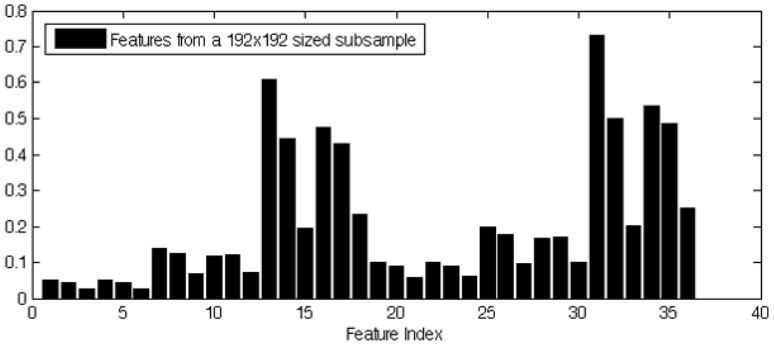
3.1.1 Example (96 × 96)



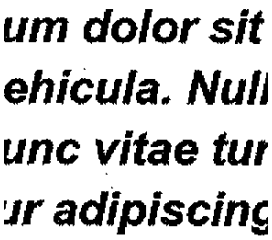
3.1.2 Features (96 × 96)



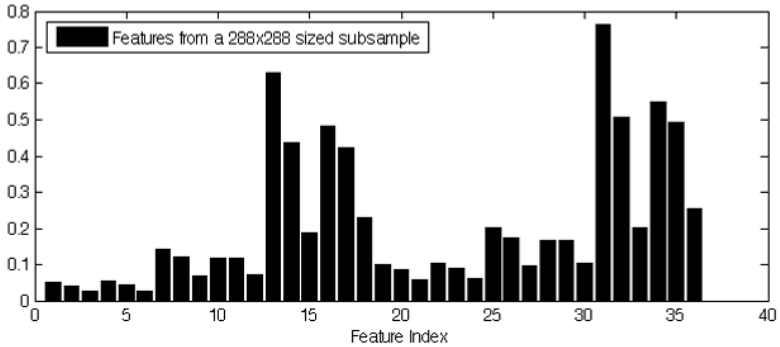
3.1.3 Example (192 × 192)



3.1.4 Features (192 × 192)



3.1.5 Example (288 × 288)



3.1.6 Features (288 × 288)

Figure 3.1: Example subsamples and average features of Arial bold, for various subsample sizes

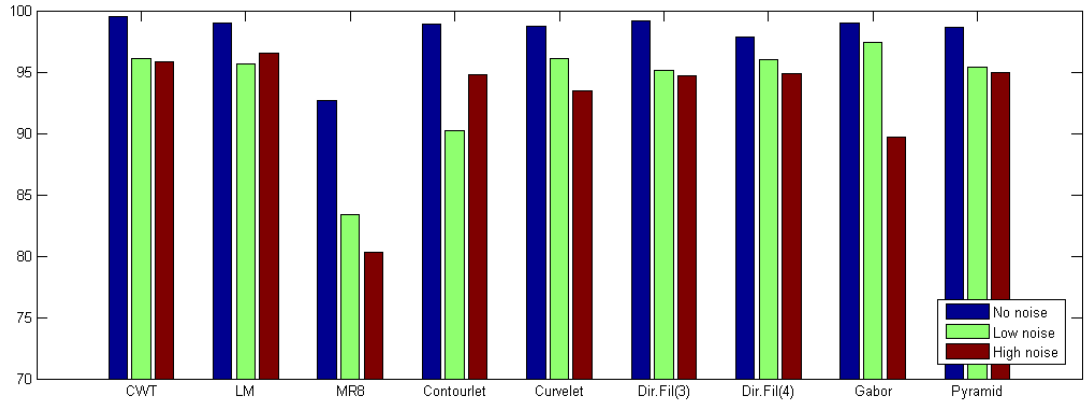


Figure 3.2: Mean recognition accuracies for each feature on datasets 1,2,3

3.2 Results on English texts

In this section we give the results of our experiments and compare the algorithms described in the introduction. In literature, Ramanathan et al. [31] use Gabor filter in a very similar fashion to our framework on this problem, so we also compare performance of [31]'s features. [31] uses Gabor filters with $\lambda = 2.7, 4.1, 5.4$ and $\theta = \pi/8, 2\pi/8, \ldots, \pi$, and in addition to taking mean and standard deviation of filter output for each filter-direction pair, they construct a maximal response image by selecting the maximum values along filter outputs for each pixel. Then mean and standard deviation of this maximal response image is appended to the feature vector. Accuracies of each feature for varying noise levels are presented in Tables 3.2-3.4.

Under no noise, all algorithms classify all font variants with very high precision. Only feature failing to surpass 97.5% level is MR8 with 92.3% mean accuracy. As noise is introduced, accuracies begin to decrease gradually. Although the effect of noise is reduced by Otsu's thresholding, it still affects the recognition process. The amount of decrease for each feature can be seen in Figure 3.2. Some features such as CWT or LM are robust to noise, and some features like Contourlet decrease sharply as noise increase.

Table 3.2: Recognition rates(%) of the feature extraction algorithms on Dataset 1 (English Printscreen texts). Emphasis of the categories denote the emphases of the class, e.g. ***A*** denotes Arial bold italic. Highest values among the algorithms is presented in bold for each case.

	Features								
Fonts	*CWT*	LM	MR8	Contourlet	Curvelet	Dir. Fil.(3)	Dir. Fil.(4)	Gabor[31]	Pyramid
A	100.00	100.00	88.89	100.00	100.00	100.00	100.00	100.00	100.00
A	100.00	92.59	81.48	96.30	92.59	100.00	88.89	96.30	92.59
A	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
A	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
B	97.22	97.22	91.67	100.00	97.22	100.00	97.22	100.00	97.22
B	97.22	97.22	88.89	100.00	100.00	100.00	97.22	100.00	100.00
B	100.00	100.00	92.59	100.00	100.00	96.30	96.30	96.30	100.00
B	100.00	100.00	94.44	97.22	100.00	97.22	97.22	100.00	100.00
CG	100.00	100.00	86.11	100.00	97.22	100.00	94.44	100.00	100.00
CG	97.22	100.00	91.67	91.67	88.89	91.67	80.56	100.00	91.67
CG	100.00	100.00	94.44	100.00	100.00	100.00	100.00	100.00	100.00
CG	100.00	94.44	75.00	94.44	100.00	100.00	100.00	100.00	97.22
CS	100.00	100.00	88.89	91.67	100.00	100.00	100.00	100.00	100.00
CS	93.33	95.56	88.89	100.00	95.56	95.56	95.56	95.56	97.78
CS	100.00	100.00	97.22	97.22	100.00	100.00	100.00	100.00	100.00
CS	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
CN	100.00	100.00	82.22	100.00	100.00	100.00	100.00	100.00	100.00
CN	100.00	100.00	71.11	100.00	100.00	100.00	97.78	97.78	100.00
CN	100.00	100.00	94.00	100.00	100.00	100.00	100.00	94.00	100.00
CN	100.00	94.00	100.00	100.00	98.00	100.00	94.00	94.00	96.00
CM	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
CM	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
CM	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
CM	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	96.30
I	100.00	100.00	92.59	100.00	100.00	100.00	100.00	100.00	100.00
I	100.00	96.30	92.59	100.00	100.00	100.00	100.00	100.00	96.30
I	100.00	100.00	96.30	100.00	100.00	100.00	100.00	100.00	100.00
I	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
T	100.00	100.00	100.00	100.00	96.30	100.00	100.00	100.00	100.00
T	100.00	100.00	88.89	96.30	96.30	96.30	96.30	92.59	92.59
T	100.00	100.00	96.30	100.00	100.00	100.00	100.00	100.00	100.00
T	100.00	100.00	92.59	100.00	96.30	96.30	96.30	100.00	100.00
Mean	99.53	98.98	92.71	98.90	98.70	99.17	97.87	98.95	98.68

Table 3.3: Recognition rates(%) of the feature extraction algorithms on Dataset 2 (English low noisy texts). Emphasis of the categories denote the emphases of the class, e.g. ***A*** denotes Arial bold italic. Highest values among the algorithms is presented in bold for each case.

	Features								
Fonts	*CWT*	LM	MR8	Contourlet	Curvelet	Dir. Fil.(3)	Dir. Fil.(4)	Gabor[31]	Pyramid
A	100.00	100.00	81.48	100.00	92.59	100.00	88.89	100.00	100.00
A	85.19	88.89	59.26	96.30	81.48	85.19	70.37	92.59	81.48
A	100.00	100.00	96.30	88.89	96.30	100.00	96.30	100.00	100.00
A	96.30	100.00	96.30	88.89	96.30	100.00	96.30	92.59	96.30
B	91.67	94.44	86.11	80.56	94.44	94.44	97.22	100.00	97.22
B	97.22	97.22	88.89	83.33	97.22	94.44	100.00	91.67	100.00
B	96.30	96.30	92.59	92.59	92.59	96.30	92.59	96.30	96.30
B	97.22	94.44	94.44	100.00	97.22	94.44	97.22	100.00	100.00
CG	100.00	97.22	80.56	88.89	91.67	97.22	91.67	94.44	94.44
CG	91.67	88.89	72.22	83.33	83.33	91.67	86.11	94.44	83.33
CG	100.00	97.22	77.78	97.22	94.44	97.22	97.22	100.00	100.00
CG	100.00	100.00	72.22	88.89	97.22	97.22	97.22	100.00	100.00
CS	97.78	97.78	66.67	88.89	95.56	100.00	100.00	100.00	100.00
CS	78.00	86.00	74.00	84.00	86.00	90.00	78.00	90.00	78.00
CS	91.67	100.00	86.11	83.33	100.00	97.22	100.00	100.00	97.22
CS	94.44	100.00	86.11	86.11	100.00	94.44	100.00	97.22	97.22
CN	100.00	97.78	57.78	82.22	100.00	95.56	100.00	100.00	97.78
CN	97.78	97.78	73.33	88.89	100.00	95.56	100.00	100.00	100.00
CN	100.00	100.00	64.44	97.78	97.78	100.00	100.00	100.00	100.00
CN	100.00	100.00	86.67	97.78	100.00	100.00	97.78	97.78	100.00
CM	100.00	96.30	96.30	100.00	100.00	100.00	100.00	100.00	100.00
CM	100.00	92.59	88.89	92.59	96.30	100.00	96.30	100.00	88.89
CM	100.00	100.00	92.59	96.30	96.30	100.00	96.30	96.30	100.00
CM	92.59	100.00	85.19	85.19	100.00	100.00	100.00	100.00	85.19
I	88.89	88.89	77.78	81.48	96.30	92.59	96.30	96.30	92.59
I	96.30	85.19	70.37	85.19	100.00	77.78	100.00	92.59	100.00
I	88.89	81.48	92.59	77.78	96.30	88.89	100.00	100.00	85.19
I	96.30	88.89	96.30	92.59	100.00	100.00	96.30	92.59	96.30
T	100.00	100.00	92.59	100.00	100.00	100.00	100.00	100.00	100.00
T	100.00	100.00	92.59	92.59	96.30	96.30	100.00	92.59	96.30
T	100.00	100.00	96.30	92.59	100.00	100.00	100.00	100.00	96.30
T	96.30	92.59	92.59	92.59	100.00	92.59	100.00	100.00	92.59
Mean	96.08	95.62	83.35	90.21	96.11	95.91	96.00	97.42	95.39

Table 3.4: Recognition rates(%) of the feature extraction algorithms on Dataset 4 (English high noisy texts). Emphasis of the categories denote the emphases of the class, e.g. ***A*** denotes Arial bold italic. Highest values among the algorithms is presented in bold for each case.

	Features								
Fonts	*CWT*	LM	MR8	Contourlet	Curvelet	Dir. Fil.(3)	Dir. Fil.(4)	Gabor[31]	Pyramid
A	100.00	100.00	88.89	100.00	88.89	100.00	100.00	85.19	96.30
A	85.19	92.59	70.37	92.59	70.37	85.19	81.48	88.89	85.19
A	100.00	100.00	96.30	100.00	96.30	100.00	100.00	96.30	100.00
A	96.30	92.59	74.07	96.30	96.30	92.59	92.59	88.89	92.59
B	100.00	97.22	77.78	91.67	97.22	100.00	97.22	91.67	97.22
B	97.22	100.00	86.11	91.67	97.22	97.22	97.22	94.44	91.67
B	88.89	92.59	96.30	96.30	85.19	92.59	88.89	92.59	96.30
B	94.44	97.22	77.78	83.33	97.22	100.00	91.67	77.78	83.33
CG	100.00	97.22	80.56	83.33	97.22	97.22	91.67	86.11	97.22
CG	80.56	94.44	83.33	86.11	72.22	83.33	97.22	72.22	80.56
CG	100.00	97.22	72.22	94.44	94.44	94.44	94.44	77.78	94.44
CG	91.67	88.89	83.33	94.44	86.11	83.33	94.44	86.11	94.44
CS	97.22	100.00	72.22	97.22	97.22	100.00	97.22	86.11	100.00
CS	97.22	100.00	80.56	100.00	97.22	97.22	100.00	88.89	100.00
CS	100.00	100.00	86.11	97.22	97.22	97.22	97.22	97.22	100.00
CS	100.00	100.00	86.11	100.00	100.00	94.44	100.00	80.56	100.00
CN	100.00	91.11	66.67	88.89	100.00	97.78	97.78	91.11	97.78
CN	88.89	91.11	64.44	86.67	84.44	88.89	88.89	82.22	86.67
CN	97.78	97.78	84.44	86.67	100.00	100.00	97.78	97.78	100.00
CN	97.78	95.56	66.67	86.67	100.00	95.56	91.11	97.78	93.33
CM	100.00	100.00	74.07	100.00	100.00	100.00	100.00	96.30	100.00
CM	100.00	100.00	70.37	100.00	96.30	100.00	100.00	96.30	100.00
CM	100.00	100.00	92.59	100.00	100.00	100.00	100.00	100.00	96.30
CM	96.30	100.00	85.19	100.00	100.00	100.00	100.00	96.30	100.00
I	93.33	95.56	82.22	100.00	100.00	95.56	95.56	95.56	97.78
I	93.33	88.89	75.56	91.11	84.44	86.67	84.44	82.22	84.44
I	97.22	97.22	88.89	100.00	100.00	100.00	97.22	94.44	97.22
I	94.44	100.00	94.44	100.00	88.89	91.67	91.67	88.89	94.44
T	96.30	100.00	81.48	96.30	100.00	100.00	100.00	96.30	100.00
T	96.30	88.89	70.37	96.30	88.89	88.89	88.89	85.19	88.89
T	85.19	96.30	66.67	96.30	85.19	81.48	88.89	85.19	96.30
T	100.00	96.30	92.59	100.00	92.59	88.89	92.59	92.59	96.30
Mean	95.80	96.52	80.27	94.80	93.47	94.69	94.88	89.65	94.96

3.3 Results on Chinese texts

Other studies on Chinese fonts include [23] where Gabor features are used and [58] in which stroke features of characters are used. Recognition rates of the tested algorithm, methods of Yang et al.[58], and Zhu et al. [23] on each font and style are presented in Table 3.5. Highest accuracy among the tested algorithms is presented in bold for each case.

Table 3.5: Recognition rates(%) of the feature extraction algorithms on Dataset 5 (Chinese texts). Emphasis of the categories denote the emphases of the class, e.g. ***FangSong*** denotes FangSong bold italic. Highest values among the algorithms is presented in bold for each case.

	Features										
Fonts	*CWT*	LM	MR8	Contourlet	Curvelet	Dir. Fil.(3)	Dir. Fil.(4)	Gabor	Pyramid	[58]	[23]
FangSong	**100.00**	93.75	87.50	87.50	**100.00**	87.50	93.75	**100.00**	87.50	92.90	**100.00**
FangSong	93.75	87.50	87.50	87.50	93.75	87.50	87.50	93.75	81.25	93.30	94.00
FangSong	**100.00**	**100.00**	87.50	93.75	**100.00**	**100.00**	93.75	**100.00**	**100.00**	91.70	96.80
FangSong	**100.00**	**100.00**	93.75	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	92.80
HeiTi	**100.00**	**100.00**	68.75	93.75	**100.00**	**100.00**	93.75	**100.00**	**100.00**	**100.00**	**100.00**
HeiTi	**100.00**	**100.00**	81.25	**100.00**	**100.00**	**100.00**	87.50	**100.00**	**100.00**	96.50	**100.00**
HeiTi	**100.00**	**100.00**	81.25	87.50	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
HeiTi	**100.00**	**100.00**	**100.00**	93.75	**100.00**	**100.00**	**100.00**	**100.00**	93.75	93.20	**100.00**
KaiTi	93.75	93.75	50.00	81.25	93.75	93.75	93.75	**100.00**	87.50	91.70	98.40
KaiTi	87.50	**100.00**	50.00	81.25	93.75	93.75	75.00	93.75	87.50	**100.00**	89.60
KaiTi	**100.00**	93.75	81.25	**100.00**	**100.00**	93.75	**100.00**	**100.00**	**100.00**	93.30	**100.00**
KaiTi	**100.00**	93.75	87.50	87.50	**100.00**	93.75	**100.00**	**100.00**	93.75	**100.00**	98.40
LiShu	**100.00**	**100.00**	68.75	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
LiShu	**100.00**	**100.00**	50.00	93.75	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
LiShu	**100.00**	**100.00**	87.50	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	96.50	**100.00**
LiShu	**100.00**	93.75	81.25	93.75	**100.00**	**100.00**	**100.00**	93.75	**100.00**	**100.00**	**100.00**
SongTi	93.75	**100.00**	81.25	93.75	87.50	93.75	93.75	**100.00**	87.50	93.20	**100.00**
SongTi	**100.00**	**100.00**	75.00	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	93.75	**100.00**	97.60
SongTi	93.75	93.75	87.50	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	99.20
SongTi	**100.00**	93.75	93.75	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	93.75	98.20	**100.00**
YouYuan	93.75	**100.00**	81.25	**100.00**	**100.00**	**100.00**	93.75	**100.00**	**100.00**	96.50	99.60
YouYuan	**100.00**	87.50	81.25	**100.00**	93.75	93.75	93.75	**100.00**	**100.00**	**100.00**	**100.00**
YouYuan	93.75	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	94.90	**100.00**
YouYuan	**100.00**	**100.00**	87.50	93.75	**100.00**	**100.00**	93.75	**100.00**	**100.00**	**100.00**	99.60
Mean	97.92	97.14	80.47	94.53	98.44	97.40	95.83	99.22	96.09	97.16	98.58

Average performance of 97.16% was reported in [58] for the six fonts. On the same set of fonts, 98,58% average performance was obtained in [23]. Several of the directional features tried in this thesis improve these rates. Gabor filters give the maximum rates among all algorithms with 99,22%.

3.4 Results on Farsi texts

Recognition accuracies for the tested algorithms on Farsi texts is comparisons with the works of [2] and [3] are provided in Table 3.6. Since recognition rates are given only for each font in [2] and [3], recognition rates of tested algorithms are also given in same manner for easy comparison. Algorithms are trained and tested with different styles for each font (consistent with other experiments in this thesis), but only font information is checked while calculating the recognition accuracy. CWT, Curvelets, Gabor filters, and Directional Filters outperform the methods in the literature. Overall, Curvelets perform best with 97.03% mean accuracy.

Table 3.6: Recognition rates(%) of the tested algorithms and comparisons with [2] and [3] on Farsi texts.

	Features										
Fonts	*CWT*	LM	MR8	Contourlet	Curvelet	Dir. Fil.(3)	Dir. Fil.(4)	Gabor	Pyramid	[2]	[3]
Homa	**100.00**	98.44	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	99.20	99.80
Lotus	92.20	89.06	85.94	84.38	**100.00**	90.63	90.63	92.19	75.00	92.20	90.70
Mitra	95.30	**100.00**	85.94	85.94	95.31	98.44	96.88	90.63	87.50	93.40	93.70
Nazanin	90.60	87.50	50.00	71.88	93.75	87.50	84.38	95.31	78.13	85.20	92.00
Tahoma	**100.00**	92.19	90.63	95.31	98.44	90.63	93.75	95.31	89.06	96.60	98.30
Times	98.40	93.75	76.56	93.75	96.88	96.88	95.31	96.88	85.94	97.20	98.80
Titr	**100.00**	**100.00**	96.88	96.88	**100.00**	96.88	**100.00**	**100.00**	**100.00**	95.20	97.00
Traffic	98.40	98.44	**100.00**	96.88	**100.00**	96.88	**100.00**	98.44	**100.00**	97.60	95.90
Yaghut	96.90	85.94	70.31	92.19	96.88	96.88	92.19	98.44	87.50	97.60	98.50
Zar	92.20	76.56	71.88	75.00	89.06	92.19	75.00	89.06	48.44	87.40	90.90
Mean	96.41	92.19	82.81	89.22	97.03	94.69	92.81	95.63	85.16	94.16	95.56

3.5 Results on Arabic texts

Table 3.7 compares the proposed method with the work of Ben Moussa et al. [4] on ALPH-REGIM dataset described in Section 1.1. Since this dataset contains images with various sizes, window size must be chosen accordingly. The smallest image in the dataset(in terms of image height) is a 100×1322 image containing 2 lines; therefore window size is chosen as 96×160, to subsample every image in dataset correctly. Almost all of tried algorithms (except LM and Contourlet) outperform the state of art.

Table 3.7: Recognition accuracies of the tested algorithms and the method of Ben Moussa et al. [4] on ALPH-REGIM database

	Features									
Fonts	*CWT*	LM	MR8	Contourlet	Curvelet	Dirfil3	Dirfil4	Gabor	Pyramid	[4]
Ahsa	99.69	98.96	0.00	98.72	99.45	99.20	98.90	99.69	99.27	94.00
Andalus	98.77	96.32	0.00	95.63	98.24	97.70	97.24	99.31	97.32	94.00
Arabictransparent	99.65	99.38	0.00	99.47	99.82	99.47	99.65	99.56	99.56	92.00
Badr	99.68	98.31	0.00	96.69	99.35	99.19	98.95	99.84	98.55	100.00
Burdiyah	97.84	98.41	0.00	94.77	97.61	97.84	98.52	97.95	98.86	100.00
Dammam	99.72	99.39	0.00	99.06	99.81	99.48	99.72	99.95	99.62	100.00
Hada	90.22	89.51	0.00	88.75	90.01	88.69	89.40	90.17	89.46	100.00
Kharj	89.88	88.53	87.13	88.12	89.06	89.06	89.29	89.99	88.77	88.00
Koufi	99.24	97.83	98.37	97.39	98.48	98.37	99.24	99.35	98.59	98.00
Naskh	98.13	96.81	90.33	92.75	97.47	96.37	97.36	98.90	96.81	98.00
Mean	97.28	96.34	27.58	95.13	96.93	96.54	96.83	97.47	96.68	96.40

3.6 Results on Ottoman texts

Different calligraphy styles have different layouts, including but not limited to the layouts shown in Figure 1.4. Although a style can be written in different layouts, page layout can be a discriminating factor of the style. In order to incorporate the layout in our features, images are subsampled as a whole page instead of using only calligraphed parts. To the best of our knowledge, Ottoman scriptures do not contain multiple calligraphy styles in a single page (unlike other printed texts used in this thesis). Therefore there is no need to divide the input image into windows; one decision per page is enough.

Table 3.8 displays the performance of tested method on the novel task of categorizing Ottoman calligraphy styles. All algorithms expect MR8 perform with over 95% accuracy.

Table 3.8: Recognition accuracies of each algorithm on Dataset 8 (Ottoman scripts)

	Calligraphy Styles					
Features	Divani	Matbu	Nesih	Rika	Talik	Mean
CWT	96.88	100.00	100.00	95.83	98.96	98.33
LM	100.00	96.88	100.00	96.88	97.92	98.33
MR8	91.67	97.92	96.88	81.25	87.50	91.04
Contourlet	95.83	98.96	95.83	87.50	97.92	95.21
Curvelet	96.88	94.79	96.88	95.83	97.92	96.46
Dir. Fil.(3)	97.92	98.96	97.92	97.92	93.75	97.29
Dir Fil.(4)	96.88	97.92	95.83	94.79	93.75	95.83
Gabor	98.96	97.92	94.79	95.83	100.00	97.50
Pyramid	93.75	98.96	97.92	94.79	97.92	96.67

Chapter 4

Experiments on Light Microscope Images

In this section we provide a detailed experimental evaluation conducted to test the effectiveness of previously mentioned algorithms on the Follicular Lymphoma grading and cancer cell line classification problems, and comparisons with related studies in the literature. In all experiments, the accuracy of the model for recognizing different fonts are computed over 10-fold cross validation. First we will establish the feature extraction framework in which the feature extraction algorithms are utilized, then we will give the results.

4.1 Feature extraction and classification framework for Follicular Lymphoma Grading problem

In Follicular Lymphoma Grading problem, we aim to grade microscope images according to their centroblast counts. Instead of counting the centroblasts individually, we try to classify the texture formed by centroblasts. Since images in this dataset are of relatively uniform texture, there is no need to segment the

images. Since it is not possible to have 2 different grades of lymphoma in an image, we only give one decision per image. Therefore there is also no need of dividing the input image to a grid like Section 3.1. We take take input image and feed it to feature extraction algorithms directly after converting to grayscale. After feature extraction, we experiment with dimension reduction. Each feature is classified once without any dimension reduction, once after principal component analysis (PCA) [59], and once after linear discriminant analysis (LDA) [60]. For PCA, dimension is reduced keeping the 99.9% of the cumulative energies of eigenvalues. For LDA, since the maximum nuber of dimensions is bounded by the number of classes, dimension is reduced to 2D for each feature.

For classification, we use the same scheme with Section 3.1.

4.2 Results on Follicular Lymphoma images

It can be seen that the directional filter features with dimensions reduced by LDA gives the best classification accuracy. M_{DirFil} is the mapping matrix of LDA for directional filtering features. The matrix is normalized to range $[-11]$, and only values with absolute value greater than 0.1 are shown.

$$M_{DirFil} = \begin{bmatrix} M_{1,\mu} \\ M_{1,\sigma} \\ M_{2,\mu} \\ M_{2,\sigma} \\ M_{3,\mu} \\ M_{3,\sigma} \end{bmatrix} \; M_{1,\mu} = \begin{bmatrix} -0.940 & -0.670 \\ 0.305 & -0.454 \\ 0.521 & -0.149 \\ 0.725 & -0.549 \\ 0.168 & 0.699 \\ 0 & 0.387 \\ -1.000 & 1.000 \\ -0.327 & -0.160 \end{bmatrix} \; M_{1,\sigma} = \begin{bmatrix} 0 & 0.162 \\ -0.187 & 0.195 \\ 0 & 0 \\ 0 & 0 \\ 0 & -0.157 \\ 0 & 0 \\ 0.128 & -0.256 \\ 0 & 0 \end{bmatrix}$$

Table 4.1: 10-fold cross-validation accuracies of each grade, for each feature

Feature	Dimension Reduction	Grades			Mean
		Grade 1	Grade 2	Grade 3	
CWT	LDA	100.00	98.88	100.00	99.63
	PCA	76.67	76.40	70.00	74.36
	None	96.67	98.88	98.89	98.14
LM	LDA	98.89	100.00	100.00	99.63
	PCA	80.00	92.13	90.00	87.38
	None	95.56	93.26	100.00	96.27
MR8	LDA	95.56	94.38	100.00	96.65
	PCA	65.56	73.03	90.00	76.20
	None	97.78	92.13	98.89	96.27
Contourlet	LDA	100.00	8.99	100.00	69.66
	PCA	82.22	82.02	84.44	82.90
	None	87.78	88.76	100.00	92.18
Curvelet	LDA	95.56	4.49	100.00	66.68
	PCA	78.89	93.26	98.89	90.35
	None	84.44	96.63	98.89	93.32
Dir. Fil(3)	LDA	100.00	100.00	100.00	100.00
	PCA	71.11	80.90	85.56	79.19
	None	97.78	97.75	98.89	98.14
Dir. Fil(4)	LDA	75.56	66.29	100.00	80.67
	PCA	52.22	100.00	100.00	84.07
	None	71.11	94.38	98.89	88.13
Gabor	LDA	0.00	0.00	100.00	33.33
	PCA	83.33	86.52	90.00	86.62
	None	86.67	89.89	95.56	90.70
GLCM	LDA	88.89	88.76	86.67	88.11
	PCA	22.22	34.83	47.78	34.94
	None	85.56	86.52	85.56	85.88
Pyramid	LDA	98.89	98.88	100.00	99.26
	PCA	71.11	94.38	67.78	77.76
	None	96.67	94.38	98.89	96.65
[35]	None	98.89	98.89	100.00	99.26

$$M_{2,\mu} = \begin{bmatrix} -0.210 & 0.282 \\ 0 & 0.224 \\ 0 & 0 \\ 0 & 0.118 \\ 0 & -0.107 \\ 0.173 & -0.285 \\ 0.148 & -0.3 \\ -0.139 & 0 \end{bmatrix} \quad M_{2,\sigma} = \begin{bmatrix} 0.120 & -0.146 \\ 0.291 & -0.171 \\ 0 & 0 \\ 0.107 & 0 \\ 0 & 0.125 \\ 0 & 0 \\ -0.346 & 0.222 \\ 0 & 0 \end{bmatrix}$$

$$M_{3,\mu} = \begin{bmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \quad M_{3,\sigma} = \begin{bmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}$$

Recall that odd and even indexed values correspond to mean and standard deviations, respectively; and features $[1, \ldots, 16]$ correspond to outputs in scale 1, features $[17, \ldots, 33]$ corresponds to scale 2, and rest correspond to scale 3. First thing we notice is that there all significant contribution comes from the first two scales. This indicates that directional information in first and second scales are discriminative enough to capture the texture completely.

Chapter 5

Experiments on Reflectance Confocal Microscopy Images

In this section we provide a detailed experimental evaluation conducted to test the effectiveness of multi-resolution directional image analysis algorithms on stratum corneum detection problem. There are no studies in the literature using computerized methods for stratum corneum detection, therefore we cannot compare performance of tested methods with the literature. As commonly done in medical area, we compare the performance of the computerized method with the manual segmentation done by an expert or experts.

5.1 Feature extraction and classification framework for Stratum Corneum thickness estimation problem

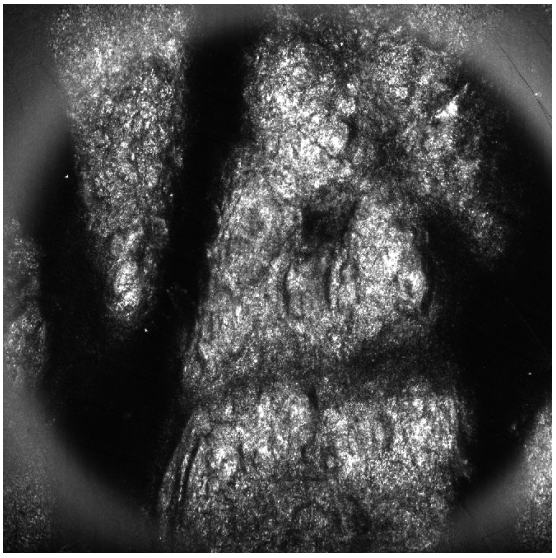
Most RCM stacks exhibit artifacts caused by reflections from the objective lens-to-skin contact window routinely used with the microscope. To reduce these artifacts and the noise, we first preprocess the images by tophat filtering followed by zero-phase filtering with a Butterworth lowpass filter. Results of this

preprocessing stage is presented in Figure 5.1.

The problem of estimating the stratum corneum thickness is equivalent to segmenting each image in a stack to SC and non-SC regions. Therefore, each image in the stack is divided into sub-windows and features are extracted from each sub-window.

For classification, we take a different approach. In Sections 3.1 and 4.1 , we used SVM, which is a supervised learning algorithm, because we had labeled data for to cover all instances of classes. This is not the case in this problem: histology of stratum layer has significant variations not only across different skin types, but across different parts of the human body as well. Therefore there is a need for large amount of data acquired from different parts of human bodies with different skin types to account for the variability of the data. Moreover, that large amount of data must be labelled by a dermatologist with expertise in RCM (who are hard to find, due to novelty of RCM in dermatology), in order to train accurate classifiers. In order to overcome these problems, we do unsupervised learning instead of supervised learning, which eliminates the need for large amount of labeled training data, reduces labor and increasing robustness across different skin types.

We cluster features into 2 classes, using k-means clustering (a standard unsupervised machine learning algorithm). Each sub-window of each image of the stack is clustered, to create a classification mask. The mask is a post-processed using median filters in order to obtain smooth clusters. Note that at this stage, although the stack is segmented into two clusters, there is no decision on which cluster corresponds to SC. This decision is made on the assumption that SC regions have higher variance than non-SC regions in top layer of the stack.



5.1.1 Before preprocessing

5.1.2 After preprocessing

Figure 5.1: Results of preprocessing for top layer of a RCM stack

5.2 Results on Reflectance Confocal Microscopy Images

For comparing results of tested methods with ground truth (GT) segmentation done by an expert, we divide the GT segmented image into same sub-windows, and calculate the thickness of stratum corneum in each sub-window. Then we calculate the average thickness difference over all sub-windows for a stack.

Table 5.1: Mean error in Stratum corneum thickness (μm) of each test stack for each feature

	CWT	Curvelet	Contourlet	GLCM	Gabor	Pyramid	Dir. Fil(3)	Dir. Fil(4)	LM	RFS
Stack 1	2.01	1.89	2.67	2.60	2.95	2.13	2.59	2.62	3.01	3.01
Stack 2	2.37	2.39	2.56	2.43	2.82	2.52	2.59	2.72	2.73	2.77
Stack 3	2.16	2.33	2.36	2.25	2.33	2.30	2.57	2.27	2.53	2.45
Stack 4	2.11	2.01	2.19	2.82	2.64	3.52	2.93	2.23	2.45	2.45
Stack 5	1.51	1.40	1.90	1.52	2.24	1.56	2.07	1.73	2.04	2.15
Stack 6	2.17	2.24	2.99	2.56	2.73	2.43	2.99	2.34	3.19	3.20
Stack 7	2.36	2.68	2.92	3.13	2.62	2.47	3.13	2.79	2.89	2.86
Mean	2.10	2.13	2.51	2.47	2.62	2.42	2.69	2.39	2.69	2.70

Chapter 6

Conclusion

A new directional filtering framework for creating oriented filters from a 1D seed filter is developed. A multi-scale filtering scheme using the directional filters is also introduced for feature extraction from images. The new multi-scale directional framework is compared with a number of multi-scale directional image representation methods including the complex wavelet transforms, curvelets, contourlets, gray level co-occurrence matrices, Gabor filters, steerable pyramids, and texton filter banks. In terms of computational efficiency, directional filter banks are the fastest among all tested methods, extracting features from a 512×512 image in 8 directions and three scales in 0.032 seconds. In terms of recognition accuracy, it is comparable to the state of the art directional feature extraction methods. Multi-scale directional filter bank is compared with other methods using an image database of font recognition in different language texts, follicular lymphoma images,and reflectance confocal microscopy images.

In font recognition problem, the CWT has the best accuracy with 99.53% average accuracy over 32 different font variants in English texts. Gabor filters outperform the CWT with 97.42% overall accuracy when some noise is introduced, and the LM filters perform the best 96.52% with mean accuracy when the noise is high. In other languages, the CWT performs best on Ottoman, and curvelets have the best performance on Persian texts with 97.03% accuracy, which is higher than the state of art reported in the literature. In Chinese documents,

the multi-scale directional filter bank, CWT, Curvelet and LM filters all have comparable font recognition results as the state of the art method reported in the literature. In this case Gabor filters outperform all of the algorithms with 99.22% accuracy. In Arabic font recognition, CWT, Curvelet, multi-scale directional filter banks, Gabor filters and scalable pyramids all outperform the state of the art. Gabor filters achieve the highest 97.47% recognition accuracy. In all of the above experiments SVM is used as the classification engine.

In follicular lymphoma grading problem, we experimented with dimension reduction. When features are reduced to 2D using linear discriminant analysis, CWT, LM filters,directional filters and steerable pyramids all perform with an accuracy above 99%, with directional filter features achieving 100%. Further inspection on mapping matrix of the LDA calculated for directional filters shows that all significant contribution comes from the first two scales, so our three-level multidirectional features are likely to be redundant for this area of application.

In Stratum corneum thickness estimation problem, features estimating the thickness closest to the expert segmentation are CWT and curvelets, with $2.1\mu m$ average error. Multi-scale directional filter bank achieved an average error of $2.39\mu m$.

After extensive performance evaluations for many features and a wide range of image analysis problems, it is clear that multi-scale directional methods are not only suitable for image representation, but for texture segmentation with a broad area of application as well. Providing results as good as the state of the art directional feature extraction methods, our multi-scale directional filter bank approach is a very computationally efficient method. It can be used in real-time applications without degrading the recognition accuracy of the state-of the art method. It is our belief that the multi-scale filter bank is a promising and alternative tool for image feature extraction and analysis.

Bibliography

[1] M. N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *Image Processing, IEEE Transactions on*, vol. 14, no. 12, pp. 2091–2106, 2005.

[2] H. Khosravi and E. Kabir, "Farsi font recognition based on sobel-roberts features," *Pattern Recognition Letters*, vol. 31, pp. 75–82, January 2010.

[3] E. M. Senobari and H. Khosravi, "Farsi font recognition based on combination of wavelet transform and sobel-robert operator features," in *Computer and Knowledge Engineering (ICCKE), 2012 2nd International eConference on*, pp. 29–33, IEEE, 2012.

[4] S. B. Moussa, A. Zahour, A. Benabdelhafid, and A. M. Alimi, "New features using fractal multi-dimensions for generalized arabic font recognition," *Pattern Recognition Letters*, vol. 31, pp. 361–371, April 2010.

[5] E. Candes, L. Demanet, D. Donoho, and L. Ying, "Fast discrete curvelet transforms," *Multiscale Modeling & Simulation*, vol. 5, no. 3, pp. 861–899, 2006.

[6] E. Le Pennec and S. Mallat, "Sparse geometric image representations with bandelets," *Image Processing, IEEE Transactions on*, vol. 14, no. 4, pp. 423–438, 2005.

[7] D. L. Donoho, "Wedgelets: Nearly minimax estimation of edges," *The Annals of Statistics*, vol. 27, no. 3, pp. 859–897, 1999.

[8] E. P. Simoncelli and W. T. Freeman, "The steerable pyramid: A flexible architecture for multi-scale derivative computation," in *Image Processing, 1995. Proceedings., International Conference on*, vol. 3, pp. 444–447, IEEE, 1995.

[9] R. A. Zuidwijk, "Directional and time-scale wavelet analysis," *SIAM Journal on Mathematical Analysis*, vol. 31, no. 2, pp. 416–430, 2000.

[10] N. Kingsbury, "Image processing with complex wavelets," *Phil. Trans. Royal Society London A*, vol. 357, pp. 2543–2560, 1997.

[11] R. H. Bamberger and M. J. Smith, "A filter bank for the directional decomposition of images: Theory and design," *Signal Processing, IEEE Transactions on*, vol. 40, no. 4, pp. 882–893, 1992.

[12] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *Systems, Man and Cybernetics, IEEE Transactions on*, no. 6, pp. 610–621, 1973.

[13] T. S. Lee, "Image representation using 2d gabor wavelets," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 10, pp. 959–971, 1996.

[14] L. Jacques, L. Duval, C. Chaux, and G. Peyré, "A panorama on multiscale geometric representations, intertwining spatial, directional and frequency selectivity," *Signal Processing*, vol. 91, no. 12, pp. 2699–2730, 2011.

[15] S. B. Moussa, A. Zahour, A. Benabdelhafid, and A. M. Alimi, "New fractal-based system for arabic/latin, printed/handwritten script identification," in *19th International Conference on Pattern Recognition (ICPR 2008)*, pp. 1–4, 2008.

[16] F. Slimane, S. Kanoun, J. Hennebert, A. M. Alimi, and R. Ingold, "A study on font-family and font-size recognition applied to arabic word images at ultra-low resolution," *Pattern Recognition Letters*, vol. 34, pp. 209–218, 2013.

[17] A. Zramdini and R. Ingold, "Optical font recognition using typographical features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 877–882, August 1998.

[18] B. Chaudhuri and U. Garain, "Automatic detection of italic, bold and all-capital words in document images," in *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, vol. 1, pp. 610–612, IEEE, 1998.

[19] J. Villegas-Cortez and C. Aviles-Cruz, "Font recognition by invariant moments of global textures," in *Proceedings of international workshop VLBV05 (very low bit-rate video-coding 2005)*, pp. 15–16, 2005.

[20] A. Amin, "Off-line arabic character recognition: the state of the art," *Pattern recognition*, vol. 31, no. 5, pp. 517–530, 1998.

[21] I. Abuhaiba, "Arabic font recognition using decision trees built from common words," *Journal of Computing and Information Technology*, vol. 13, no. 3, pp. 211–224, 2004.

[22] A. Borji and M. Hamidi, "Support vector machine for persian font recognition," *International Journal of Intelligent Systems and Technologies*, pp. 184–187, 2007.

[23] Y. Zhu, T. Tan, and Y. Wang, "Font recognition based on global texture analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence - Graph Algorithms and Computer Vision*, vol. 23, pp. 1192–1200, October 2001.

[24] H. Ma and D. Doermann, "Gabor filter based multi-class classifier for scanned document images," in *7th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 968 – 972, 2003/// 2003.

[25] C. Aviles-Cruz, R. Rangel-Kuoppa, M. Reyes-Ayala, A. Andrade-Gonzalez, and R. Escarela-Perez, "High-order statistical texture analysis-font recognition applied," *Pattern Recognition Letters*, vol. 26, pp. 135–145, January 2005.

[26] E. Rashedi, H. Nezamabadi-pour, and S. Saryzadi, "Farsi font recognition using correlation coefficients (in farsi)," in *4th Conf. on Machine Vision and Image Processing, Ferdosi Mashhad*, 2007.

[27] F. Aiolli, M. Simi, D. Sona, A. Sperduti, A. Starita, and G. Zaccagnini, "Spi: A system for paleographic inspections," in *AI*IA Notizie*, 1999.

[28] I. B. Yosef, I. Beckman, K. Kedem, and I. Dinstein, "Binarization, character extraction, and writer identification of historical hebrew calligraphy documents," *International Journal on Document Analysis and Recognition*, vol. 9, no. 2-4, pp. 89–99, 2007.

[29] Y. Zhuang, W. Lu, and J. Wu, "Latent style model: Discovering writing styles for calligraphy works," *Journal of Visual Communication and Image Representation*, vol. 20, pp. 84–96, 2009.

[30] M. S. Azmi, K. Omar, M. F. Nasrudin, A. K. Muda, and A. Abdullah, "Arabic calligraphy classification using triangle model for digital jawi paleography analysis," in *11th International Conference on Hybrid Intelligent Systems (HIS)*, pp. 704–708, 2011.

[31] R. Ramanathan, K. Soman, L. Thaneshwaran, V. Viknesh, T. Arunkumar, and P. Yuvaraj, "A novel technique for english font recognition using support vector machines," in *International Conference on Advances in Recent Technologies in Communication and Computing(ARTCom '09)*, pp. 766–769, 2009.

[32] E. F. Can, F. Can, P. Duygulu, and M. Kalpakli, "Automatic categorization of ottoman literary texts by poet and time period," in *Computer and Information Sciences II*, pp. 51–57, Springer, 2012.

[33] M. Garcia, A. Jemal, E. Ward, M. Center, Y. Hao, R. Siegel, and M. Thun, "Global cancer facts & figures 2007," *Atlanta, GA: American Cancer Society*, vol. 1, no. 3, 2007.

[34] E. S. Jaffe, "The 2008 who classification of lymphomas: implications for clinical practice and translational research," *ASH Education Program Book*, vol. 2009, no. 1, pp. 523–531, 2009.

[35] A. Suhre, *Novel Methods for Microscopic Image Processing, Analysis, Classification and Compression.* PhD thesis, Bilkent University, 2013.

[36] M. Rajadhyaksha, M. Grossman, D. Esterowitz, R. H. Webb, and R. R. Anderson, "In vivo confocal scanning laser microscopy of human skin: melanin provides strong contrast," *Journal of Investigative Dermatology*, vol. 104, no. 6, pp. 946–952, 1995.

[37] G. T. Herman and A. Kuba, *Discrete tomography: Foundations, algorithms, and applications.* Springer, 1999.

[38] M. J. Shensa, "The discrete wavelet transform: wedding the a trous and mallat algorithms," *Signal Processing, IEEE Transactions on*, vol. 40, no. 10, pp. 2464–2482, 1992.

[39] N. Kingsbury, "A dual-tree complex wavelet transform with improved orthogonality and symmetry properties," in *Image Processing, 2000. Proceedings. 2000 International Conference on*, vol. 2, pp. 375–378, IEEE, 2000.

[40] N. Kingsbury, "The dual-tree complex wavelet transform: a new efficient tool for image restoration and enhancement," in *Proc. EUSIPCO*, vol. 98, pp. 319–322, 1998.

[41] I. Selesnick, R. Baraniuk, and N. Kingsbury, "The dual-tree complex wavelet transform," *Signal Processing Magazine, IEEE*, vol. 22, no. 6, pp. 123–151, 2005.

[42] A. F. Abdelnour and I. W. Selesnick, "Nearly symmetric orthogonal wavelet bases," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, vol. 6, 2001.

[43] E. Candes and D. Donoho, "Curvelets: A surprisingly effective nonadaptive representation for objects with edges," 2000.

[44] E. J. Candes and D. L. Donoho, "New tight frames of curvelets and optimal representations of objects with piecewise c2 singularities," *Communications on pure and applied mathematics*, vol. 57, no. 2, pp. 219–266, 2004.

[45] E. J. Candès and D. L. Donoho, "Ridgelets: A key to higher-dimensional intermittency?," *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 357, no. 1760, pp. 2495–2509, 1999.

[46] S.-M. Phoong, C. W. Kim, P. Vaidyanathan, and R. Ansari, "A new class of two-channel biorthogonal filter banks and wavelet bases," *Signal Processing, IEEE Transactions on*, vol. 43, no. 3, pp. 649–665, 1995.

[47] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Transactions on Pattern analysis and machine intelligence*, vol. 13, no. 9, pp. 891–906, 1991.

[48] A. Karasaridis and E. P. Simoncelli, "A filter design technique for steerable pyramid image transforms," in *ICASSP*, (Atlanta, GA), May 1996.

[49] B. Julesz, "Textons, the elements of texture perception, and their interactions.," *Nature*, 1981.

[50] J. Malik, S. Belongie, J. Shi, and T. Leung, "Textons, contours and regions: Cue integration in image segmentation," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, pp. 918–925, IEEE, 1999.

[51] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, 2001.

[52] C. Schmid, "Constructing models for content-based image retrieval," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 2, pp. II–39, IEEE, 2001.

[53] J. M. Geusebroek, A. W. M. Smeulders, and J. van de Weijer, "Fast anisotropic gauss filtering," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 938–943, 2003.

[54] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Optical Society of America, Journal, A: Optics and Image Science*, vol. 2, no. 7, pp. 1160–1169, 1985.

[55] N. Petkov and P. Kruizinga, "Computational models of visual neurons specialised in the detection of periodic and aperiodic oriented visual stimuli: bar and grating cells," *Biological cybernetics*, vol. 76, no. 2, pp. 83–96, 1997.

[56] J. P. Jones and L. A. Palmer, "An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex," *Journal of Neurophysiology*, vol. 58, no. 6, pp. 1233–1258, 1987.

[57] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[58] Z. Yang, L. Yang, D. Qi, and C. Suen, "An emd-based recognition method for chinese fonts and styles," *Pattern Recognition Letters*, vol. 27, pp. 1692–1701, October 2006.

[59] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2005.

[60] C. R. Rao, "The utilization of multiple measurements in problems of biological classification," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 10, no. 2, pp. 159–203, 1948.

Appendix A

Confusion Matrices of optimal classifiers in Font Recognition Problem

Table A.1: Confusion matrix of the optimal classifier trained with CWT features in no-noise set (Dataset 1)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	97.22	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	2.22	0	0	0	0	0	0	0	4.44	0	0	0	93.33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table A.2: Confusion matrix of the optimal classifier trained with contourlet features in no-noise set (Dataset 1)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	96.30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	2.78	0	0	0	91.67	0	0	0	5.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	2.78	0	0	2.78	94.44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	91.67	0	8.33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table A.3: Confusion matrix of the optimal classifier trained with curvelet features in no-noise set (Dataset 1)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	92.59	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	97.22	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	97.22	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	8.33	0	0	0	2.78	0	0	0	88.89	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	4.44	0	0	0	95.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.00	98.00	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	96.30	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	96.30	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
T	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30

Table A.4: Confusion matrix of the optimal classifier trained with dirfil3 features in no-noise set (Dataset 1)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	96.30	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78
CG	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	2.78	0	0	0	0	0	0	0	91.67	0	0	0	5.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	2.22	0	0	0	2.22	0	0	0	95.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
T	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30

Table A.5: Confusion matrix of the optimal classifier trained with dirfil4 features in no-noise set (Dataset 1)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	88.89	0	0	0	0	0	0	0	11.11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	97.22	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	97.22	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	96.30	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78
CG	0	0	0	0	0	0	0	0	94.44	0	0	0	0	2.78	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	8.33	0	0	0	0	0	0	0	80.56	0	0	0	11.11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	2.22	0	0	0	2.22	0	0	0	0	95.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.22	97.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.00	94.00	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	96.30	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
T	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30

Table A.6: Confusion matrix of the optimal classifier trained with gabor features in no-noise set (Dataset 1)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	96.30	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	3.70	0	96.30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	2.22	0	0	0	0	0	0	0	2.22	0	0	0	95.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	0	0	97.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	2.00	0	0	0	0	0	0	0	0	0	0	0	4.00	0	94.00	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	4.00	0	0	0	2.00	0	94.00	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	3.70	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	92.59	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table A.7: Confusion matrix of the optimal classifier trained with LM features in no-noise set (Dataset 1)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	92.59	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	97.22	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	5.56	0	0	0	94.44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	2.22	0	0	0	0	0	0	0	2.22	0	0	0	95.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.00	94.00	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	0	0	3.70	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table A.8: Confusion matrix of the optimal classifier trained with MR8 features in no-noise set (Dataset 1)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	88.89	3.70	0	0	0	0	0	0	3.70	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	11.11	81.48	0	0	0	0	0	0	0	7.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	2.78	0	0	0	91.67	2.78	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	8.33	88.89	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	92.59	7.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	2.78	94.44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78
CG	2.78	2.78	0	0	0	0	0	0	86.11	5.56	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	8.33	91.67	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	94.44	5.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	25.00	75.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	2.78	2.78	0	0	88.89	5.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	2.22	0	0	0	0	0	0	0	8.89	88.89	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97.22	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	82.22	17.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28.89	71.11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94.00	6.00	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	92.59	7.41	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.41	92.59	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	0	0	0	3.70
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	3.70	0	0	0	3.70	88.89	0	0
T	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0
T	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	92.59

Table A.9: Confusion matrix of the optimal classifier trained with pyramid features in no-noise set (Dataset 1)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	92.59	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	5.56	0	0	0	0	0	0	0	91.67	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	2.78	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	2.22	0	0	0	97.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4.00	96.00	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	96.30	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	0	0	3.70	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.41	0	0	0	92.59	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table A.10: Confusion matrix of the optimal classifier trained with CWT features in low-noise set (Dataset 2)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	85.19	0	0	0	0	0	0	0	0	0	0	0	14.81	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	96.30	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	91.67	0	0	0	0	0	0	0	0	0	0	0	8.33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	96.30	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	5.56	0	0	0	0	0	0	0	91.67	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	97.78	0	2.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	6.00	0	0	6.00	0	0	0	0	6.00	0	0	0	78.00	0	0	0	4.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	8.33	0	91.67	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	2.78	0	0	0	0	0	0	0	0	0	2.78	0	94.44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	0	0	97.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.41	0	92.59	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88.89	0	11.11	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	3.70	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11.11	0	88.89	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	96.30	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
T	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30

Table A.11: Confusion matrix of the optimal classifier trained with contourlet features in low-noise set (Dataset 2)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	96.30	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	88.89	0	0	0	11.11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	3.70	88.89	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.41	0	0
B	0	0	0	0	80.56	2.78	2.78	0	2.78	0	0	0	0	0	0	0	11.11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	2.78	83.33	0	0	5.56	0	0	0	0	0	0	0	2.78	5.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	3.70	0	92.59	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	2.78	5.56	0	0	88.89	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	2.78	0	0	0	2.78	0	0	2.78	83.33	0	0	0	5.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0
CG	0	0	0	0	0	0	0	0	2.78	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	8.33	0	0	2.78	88.89	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	88.89	0	11.11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	2.00	0	0	0	0	0	0	0	2.00	2.00	0	0	84.00	0	6.00	0	0	2.00	2.00	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	16.67	0	83.33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	13.89	0	86.11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	2.22	2.22	0	0	2.22	0	0	0	0	0	0	0	82.22	8.89	2.22	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	0	8.89	88.89	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.22	0	97.78	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.22	0	97.78	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	92.59	0	7.41	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	96.30	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14.81	0	85.19	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	81.48	0	18.52	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	85.19	0	11.11	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22.22	0	77.78	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.41	0	92.59	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	92.59	0	7.41
T	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	92.59	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.41	0	92.59

Table A.12: Confusion matrix of the optimal classifier trained with curvelet features in low-noise set (Dataset 2)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	92.59	0	0	0	0	0	0	0	7.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	81.48	0	0	0	0	0	0	0	11.11	0	0	0	7.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	96.30	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	96.30	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	2.78	0	0	0	94.44	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	2.78	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	92.59	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	97.22	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	5.56	0	0	0	0	0	0	0	91.67	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	13.89	0	0	0	2.78	0	0	0	83.33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	94.44	0	0	0	0	0	0	0	5.56	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	2.78	0	0	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	2.22	0	0	0	0	0	0	0	0	0	0	0	95.56	0	0	0	2.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	4.00	0	0	2.00	0	0	0	0	0	0	2.00	0	86.00	0	6.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	2.22	0	0	0	0	0	0	0	97.78	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	3.70	0	0	0	0	0	0	0	0
CM	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	3.70	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	96.30	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table A.13: Confusion matrix of the optimal classifier trained with dirfil3 features in low-noise set (Dataset 2)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	85.19	0	0	0	0	0	0	0	7.41	0	0	0	7.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	94.44	0	0	0	0	0	0	0	2.78	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	94.44	0	0	0	0	0	0	0	0	0	0	0	5.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	96.30	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	94.44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	2.78
CG	0	0	0	0	0	0	0	0	97.22	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	5.56	0	0	0	0	0	0	0	91.67	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	97.22	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	0	97.22	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	2.00	0	0	0	2.00	0	0	6.00	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	94.44	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	0	2.22	95.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	4.44	0	0	0	0	0	0	0	0	0	0	0	95.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	92.59	0	7.41	0	0	0	0	0
I	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	77.78	0	14.81	0	3.70	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11.11	0	88.89	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	3.70
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	3.70	0	0	0	92.59

Table A.14: Confusion matrix of the optimal classifier trained with dirfil4 features in low-noise set (Dataset 2)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	88.89	0	0	0	7.41	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	70.37	0	0	0	0	0	0	0	18.52	0	0	0	11.11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	96.30	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	96.30	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	97.22	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	3.70	0	0	0	92.59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70
B	0	0	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78
CG	2.78	0	0	0	0	0	0	0	91.67	2.78	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	8.33	0	0	0	0	0	0	0	86.11	0	0	0	5.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	97.22	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	0	97.22	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	2.00	0	0	0	0	0	0	4.00	6.00	0	0	4.00	78.00	0	0	0	4.00	0	2.00	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	0	0	97.78	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	3.70	0	0	0	0	0	0	0	0
CM	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	0	0	3.70	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
I	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table A.15: Confusion matrix of the optimal classifier trained with gabor features in low-noise set (Dataset 2)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	92.59	0	0	0	3.70	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	92.59	0	7.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	8.33	0	0	0	91.67	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	3.70	0	96.30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	2.78	0	0	0	2.78	0	0	94.44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	2.78	0	0	0	0	0	0	0	94.44	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	4.00	0	0	2.00	0	0	0	0	0	0	0	2.00	90	0	2.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	2.22	0	0	0	0	0	0	0	97.78	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	96.30	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	0	0	3.70	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	92.59	0	3.70	0	3.70	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
I	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	92.59	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	92.59	0	3.70
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table A.16: Confusion matrix of the optimal classifier trained with LM features in low-noise set (Dataset 2)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	88.89	0	0	0	0	0	0	0	7.41	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	94.44	0	0	0	0	0	0	0	0	2.78	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	96.30	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	94.44	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78
CG	0	0	0	0	0	0	0	0	97.22	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	8.33	0	0	0	0	0	0	0	88.89	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	97.22	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	97.78	0	2.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	6.00	0	0	0	0	6.00	0	0	0	86.00	0	0	0	0	0	2.00	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	0	0	97.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97.78	0	0	0	0	0	0	0	0	0	0	0	2.22	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	3.70	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	92.59	0	7.41	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88.89	0	11.11	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	85.19	0	14.81	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18.52	0	81.48	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	88.89	0	0	0	7.41
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
T	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	92.59

Table A.17: Confusion matrix of the optimal classifier trained with MR8 features in low-noise set (Dataset 2)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	81.48	3.70	0	0	11.11	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	11.11	59.26	0	0	0	0	0	0	11.11	14.81	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	96.30	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	3.70	96.30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	2.78	0	0	0	86.11	8.33	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	11.11	88.89	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	7.41	0	0	0	92.59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	94.44	0	0	0	0	0	0	0	0	0	0	5.56	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	5.56	0	0	0	0	0	0	80.56	8.33	0	0	0	5.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	8.33	0	0	0	0	0	0	13.89	72.22	0	0	2.78	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	77.78	22.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	25.00	72.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78
CS	0	4.44	0	0	0	2.22	0	0	0	0	0	0	66.67	26.67	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	2.00	0	0	0	2.00	0	0	0	20	74.00	0	0	2.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	86.11	5.56	2.78	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13.89	86.11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	2.22	0	0	57.78	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4.44	2.22	20	73.33	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	8.89	0	0	0	0	0	0	0	0	0	0	64.44	22.22	0	0	0	0	0	0	0	0	0	0	2.22	2.22
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13.33	86.67	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	3.70	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	88.89	0	7.41	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	92.59	3.70	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11.11	3.70	85.19	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	77.78	18.52	0	0	3.70	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29.63	70.37	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	92.59	7.41	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	96.30	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	3.70	0	0	0	92.59	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.41	0	0	0	0	0	0	0	0	0	0	0	0	92.59	0	0
T	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0
T	0	0	0	0	0	0	3.70	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	92.59

Table A.18: Confusion matrix of the optimal classifier trained with pyramid features in low-noise set (Dataset 2)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	81.48	0	0	0	0	0	0	0	14.81	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	96.30	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	96.30	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	2.78	0	0	0	94.44	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	11.11	0	0	0	0	0	0	0	83.33	0	0	0	5.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	2.00	0	0	0	0	0	2.00	2.00	6.00	0	0	0	78.00	0	2.00	2.00	2.00	4.00	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	0	0	97.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88.89	0	11.11	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14.81	0	85.19	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	92.59	0	7.41	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14.81	0	85.19	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	0	0	3.70
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	96.30	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	3.70	0	0	0	92.59

Table A.19: Confusion matrix of the optimal classifier trained with CWT features in high-noise set (Dataset 3)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	85.19	0	0	0	0	0	0	0	11.11	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	96.30	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	2.78	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	88.89	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.41	0	3.70	0	0	0	0	0
B	0	0	0	0	0	0	0	94.44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	2.78	0
CG	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	8.33	0	0	0	0	0	0	0	80.56	0	0	0	8.33	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	8.33	0	0	0	91.67	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	97.22	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	2.78	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88.89	0	2.22	0	0	0	0	0	8.89	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.22	0	97.78	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.22	0	97.78	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	96.30	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	4.44	0	0	0	0	0	0	0	0	0	0	2.22	0	0	0	0	0	0	93.33	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	0	4.44	0	0	0	0	0	0	93.33	0	0	0	0	0	0
I	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97.22	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	94.44	0	0	2.78	0
T	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	96.30	0	0
T	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11.11	0	0	85.19	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table A.20: Confusion matrix of the optimal classifier trained with contourlet features in high-noise set (Dataset 3)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	92.59	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	96.30	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	91.67	5.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0
B	0	0	0	0	5.56	91.67	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	96.30	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	83.33	0	0	8.33	8.33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	2.78	0	0	0	2.78	2.78	0	0	83.33	2.78	5.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	2.78	0	0	0	2.78	0	0	0	86.11	0	0	0	2.78	0	2.78	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	2.78	0	94.44	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	2.78	0	94.44	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	97.22	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88.89	0	11.11	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	86.67	0	13.33	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13.33	0	86.67	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11.11	2.22	86.67	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	2.22	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	0	2.22	91.11	0	2.22	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	0	3.70
T	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	0
T	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table A.21: Confusion matrix of the optimal classifier trained with curvelet features in high-noise set (Dataset 3)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	88.89	0	0	0	0	0	0	0	7.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0
A	0	70.37	0	0	0	0	0	0	0	22.22	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0
A	0	0	96.30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70
A	0	0	0	96.30	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	97.22	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0
B	0	0	0	0	0	0	85.19	0	0	0	11.11	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	97.22	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	2.78	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	16.67	0	0	0	8.33	0	0	0	72.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0
CG	0	0	0	0	0	0	0	0	0	0	94.44	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	2.78	0	0	0	0	0
CG	0	0	0	8.33	0	0	0	5.56	0	0	0	86.11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	2.78	0	0	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	2.78	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	84.44	0	0	0	0	0	0	0	13.33	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	96.30	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15.56	0	0	0	0	0	0	0	84.44	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
I	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	88.89	0	0	5.56	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	3.70	0	0	0	0	0	0	7.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88.89	0	0
T	0	0	0	3.70	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.41	0	0	85.19	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.41	0	0	0	0	92.59

Table A.22: Confusion matrix of the optimal classifier trained with dirfil3 features in high-noise set (Dataset 3)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	85.19	0	0	0	0	0	0	0	7.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.41	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	92.59	0	0	0	0	0	0	0	7.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	97.22	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	92.59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	3.70	0	0	0	0	0
B	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	97.22	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	2.78	0	2.78	0	2.78	0	0	0	83.33	0	0	0	8.33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	94.44	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0
CG	0	0	0	11.11	0	0	0	2.78	0	0	2.78	83.33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	2.78	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	2.78	0	0	0	0	0	0	0	0	0	0	0	2.78	0	94.44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	97.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	88.89	0	0	0	0	0	0	0	6.67	0	2.22	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	0	0	95.56	0	0	0	0	0	0	0	2.22	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95.56	0	0	0	2.22	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13.33	0	0	0	0	0	0	0	86.67	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	2.78	0	0	0	0	0	0	0	91.67	0	0	2.78	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.41	0	0	0	88.89	0	0
T	0	0	0	3.70	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11.11	0	0	81.48	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11.11	0	0	0	0	88.89

Table A.23: Confusion matrix of the optimal classifier trained with dirfil4 features in high-noise set (Dataset 3)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	81.48	0	0	0	0	0	0	7.41	11.11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	92.59	0	0	0	0	0	0	0	7.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	97.22	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0
B	0	0	0	0	0	0	88.89	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	7.41	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	91.67	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	5.56	0
CG	2.78	0	0	0	0	0	0	0	91.67	0	0	0	0	0	0	5.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	2.78	0	0	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	2.78	0	0	0	0	0	0	0	94.44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0
CG	0	0	0	2.78	0	0	0	0	0	0	2.78	94.44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	97.22	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97.78	0	0	0	0	0	0	0	0	0	2.22	0	0	0	0	0
CN	0	0	0	0	0	0	0	2.22	2.22	0	0	0	0	0	0	0	0	88.89	0	0	0	0	0	0	0	6.67	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	2.22	0	0	0	0	0	0	0	97.78	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	2.22	0	91.11	0	0	0	0	0	0	0	4.44	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	4.44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95.56	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	13.33	0	0	0	0	0	0	0	84.44	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	97.22	0	0	0	0	0
I	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	91.67	0	0	2.78	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	0	0	0	3.70	7.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88.89	0	0
T	0	0	0	0	0	0	0	7.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	88.89	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.41	0	0	0	0	92.59

Table A.24: Confusion matrix of the optimal classifier trained with gabor features in high-noise set (Dataset 3)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	85.19	0	0	0	3.70	3.70	0	0	7.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	88.89	0	0	0	0	0	0	7.41	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	96.30	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	88.89	0	0	0	0	0	0	7.41	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	91.67	2.78	0	0	0	0	0	0	5.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	2.78	0	0	0	0	94.44	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	92.59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	3.70	0	0	0	0	0
B	0	0	0	0	0	0	0	77.78	0	0	5.56	8.33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	5.56	0
CG	2.78	5.56	0	0	0	5.56	0	0	86.11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	13.89	0	0	0	2.78	0	0	5.56	72.22	0	0	0	2.78	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0
CG	0	0	0	2.78	0	0	0	8.33	0	0	77.78	5.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	2.78	0	0	0	0
CG	0	0	0	2.78	0	0	0	2.78	0	0	5.56	86.11	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0
CS	2.78	0	0	0	5.56	0	0	0	0	0	0	0	86.11	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0
CS	0	0	0	0	0	5.56	0	0	2.78	0	0	0	0	88.89	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0
CS	0	2.78	0	0	0	2.78	0	0	0	11.11	0	0	0	2.78	0	80.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	91.11	0	2.22	0	0	0	0	0	0	6.67	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4.44	82.22	0	0	0	0	0	0	0	11.11	0	0	0	2.22	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.22	0	97.78	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	97.78	0	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	0	0	0	3.70	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	96.30	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	96.30	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4.44	0	0	0	0	0	0	0	0	0	95.56	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	2.22	0	0	2.22	8.89	0	2.22	0	0	0	0	0	82.22	0	0	0	2.22	0	0
I	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	94.44	0	0	0	0	0
I	0	0	0	0	0	0	0	8.33	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	88.89	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30	0	0	0
T	0	0	0	0	0	0	0	0	0	3.70	0	0	0	11.11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	85.19	0	0
T	0	0	0	0	0	0	0	0	0	0	0	7.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.41	0	0	85.19	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.41	0	0	0	0	92.59

Table A.25: Confusion matrix of the optimal classifier trained with LM features in high-noise set (Dataset 3)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	92.59	0	0	0	0	0	0	0	7.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	92.59	0	0	0	0	0	0	0	7.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0
B	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	92.59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.41	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	97.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0
CG	0	0	0	0	0	0	0	0	97.22	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	5.56	0	0	0	0	0	0	0	94.44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	0	0	0	0	0	0	0	97.22	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	8.33	0	0	0	0	0	0	2.78	88.89	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	2.22	0	0	0	0	0	91.11	0	6.67	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	91.11	0	2.22	0	0	0	0	0	4.44	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.22	0	97.78	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	0	0	95.56	0	0	0	0	0	0	0	2.22	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
I	0	0	2.22	0	0	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	0	0	0	0	95.56	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.67	0	0	0	0	0	0	0	88.89	0	0	0	4.44	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97.22	0	0	0	0	2.78
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.41	0	0	0	88.89	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	96.30	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.30

Table A.26: Confusion matrix of the optimal classifier trained with MR8 features in high-noise set (Dataset 3)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	88.89	3.70	0	0	0	0	0	0	3.70	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	7.41	70.37	0	0	0	3.70	0	0	3.70	14.81	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	96.30	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	7.41	74.07	0	0	0	3.70	0	0	7.41	7.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	77.78	16.67	0	0	0	0	0	0	2.78	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	2.78	0	0	11.11	86.11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	96.30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0
B	0	0	0	2.78	0	0	0	77.78	0	0	2.78	0	0	0	0	0	2.78	5.56	0	2.78	0	0	0	0	2.78	0	0	0	0	0	2.78	0
CG	0	8.33	0	0	0	0	0	0	80.56	5.56	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0
CG	5.56	5.56	0	0	0	2.78	0	0	0	83.33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0
CG	0	0	8.33	5.56	0	0	0	2.78	0	0	72.22	11.11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	2.78	2.78	0	0	0	0	0	0	11.11	83.33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	2.78	0	0	0	0	0	0	0	0	0	0	0	72.22	25.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	19.44	80.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	86.11	2.78	0	0	0	0	0	0	0	0	8.33	2.78	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11.11	86.11	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0
CN	0	0	0	2.22	0	0	0	6.67	0	0	0	0	0	0	0	0	66.67	17.78	0	2.22	0	0	0	0	2.22	0	0	0	0	0	0	2.22
CN	0	0	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	24.44	64.44	0	4.44	0	0	0	0	0	2.22	0	0	0	0	0	2.22
CN	0	0	2.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	84.44	13.33	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	2.22	0	0	0	0	0	0	0	0	0	0	0	0	0	2.22	0	22.22	66.67	0	0	0	0	0	0	2.22	0	0	0	4.44	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	74.07	25.93	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29.63	70.37	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	92.59	7.41	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	11.11	85.19	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	6.67	0	0	0	0	0	0	0	82.22	6.67	2.22	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8.89	0	0	2.22	0	0	0	0	0	0	13.33	75.56	0	0	0	0	0	0
I	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	88.89	2.78	0	0	0	2.78
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	2.78	94.44	0	0	0	0
T	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	81.48	11.11	0	0
T	0	0	0	0	3.70	3.70	0	0	0	7.41	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	11.11	70.37	0	0
T	0	0	0	0	0	0	0	22.22	0	0	0	0	0	0	0	0	3.70	0	0	7.41	0	0	0	0	0	0	0	0	0	0	66.67	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.41	0	0	0	0	92.59

Table A.27: Confusion matrix of the optimal classifier trained with pyramid features in high-noise set (Dataset 3)

	A	***A***	*A*	A	**B**	***B***	*B*	B	**CG**	***CG***	*CG*	CG	**CS**	***CS***	*CS*	CS	**CN**	***CN***	*CN*	CN	**CM**	***CM***	*CM*	CM	**I**	***I***	*I*	I	**T**	***T***	*T*	T
A	96.30	0	0	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	85.19	0	0	0	0	0	0	0	11.11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0
A	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	92.59	0	0	0	0	0	0	0	7.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	97.22	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	91.67	0	0	0	5.56	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	96.30	0	0	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	83.33	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	2.78	0	8.33	0	2.78	0	0
CG	0	0	0	0	0	0	0	0	97.22	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	11.11	0	0	0	5.56	0	0	0	80.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0
CG	0	0	0	0	0	0	2.78	0	0	0	94.44	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CG	0	0	0	5.56	0	0	0	0	0	0	0	94.44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97.78	0	2.22	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	4.44	0	0	0	0	0	0	0	0	0	86.67	0	0	0	0	0	0	0	6.67	0	2.22	0	0	0	0
CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
CN	0	0	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	0	0	93.33	0	0	0	0	0	0	0	4.44	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	0	0	0	96.30	0	0	0	0	0	0	0	0	0
CM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97.78	0	2.22	0	0	0	0	0
I	0	2.22	0	0	0	0	0	2.22	0	0	0	0	0	0	0	0	0	11.11	0	0	0	0	0	0	0	84.44	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97.22	0	0	0	0	2.78
I	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	0	0	0	0	2.78	0	0	0	0	0	0	0	94.44	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
T	0	0	0	0	0	0	0	3.70	0	3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	88.89	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	96.30	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.70	0	0	0	0	96.30

Table A.28: Confusion matrix of the optimal classifier trained with CWT features in Chinese dataset

	FS	***FS***	*FS*	FS	**HT**	***HT***	*HT*	HT	**KT**	***KT***	*KT*	KT	**LS**	***LS***	*LS*	LS	**ST**	***ST***	*ST*	ST	**YY**	***YY***	*YY*	YY
FS	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	93.75	0	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	93.75	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	6.25	0	0	6.25	87.50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.25	0	93.75	0	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
ST	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.75	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
YY	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	93.75	0	0	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
YY	0	0	0	0	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	93.75	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table A.29: Confusion matrix of the optimal classifier trained with contourlet features in Chinese dataset

	FS	***FS***	*FS*	FS	**HT**	***HT***	*HT*	HT	**KT**	***KT***	*KT*	KT	**LS**	***LS***	*LS*	LS	**ST**	***ST***	*ST*	ST	**YY**	***YY***	*YY*	YY
FS	87.50	0	0	0	0	0	0	0	6.25	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	87.50	0	0	0	0	0	0	0	6.25	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	0	93.75	0	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	6.25	0	0	0	93.75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	6.25	0	87.50	0	0	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	0	0	93.75	0	0	0	0	0	0	0	0	0	0	0	0	0	6.25	0	0
KT	0	0	0	0	0	0	0	0	81.25	12.50	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	6.25	0	0	0	0	0	0	0	81.25	0	12.50	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	6.25	0	0	0	0	0	0	6.25	87.50	0	0	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	6.25	93.75	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	6.25	0	93.75	0	0	0	0	0	0	0	0
ST	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.75	0	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
YY	0	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.75

Table A.30: Confusion matrix of the optimal classifier trained with curvelet features in Chinese dataset

	FS	***FS***	*FS*	FS	**HT**	***HT***	*HT*	HT	**KT**	***KT***	*KT*	KT	**LS**	***LS***	*LS*	LS	**ST**	***ST***	*ST*	ST	**YY**	***YY***	*YY*	YY
FS	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	93.75	0	0	0	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	6.25	0	0	0	0	0	0	0	93.75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	6.25	93.75	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	6.25	0	0	0	0	0	87.50	0	0	0	6.25	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.25	0	0	0	0	0	93.75	0	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table A.31: Confusion matrix of the optimal classifier trained with dirfil3 features in Chinese dataset

	FS	***FS***	*FS*	FS	**HT**	***HT***	*HT*	HT	**KT**	***KT***	*KT*	KT	**LS**	***LS***	*LS*	LS	**ST**	***ST***	*ST*	ST	**YY**	***YY***	*YY*	YY
FS	87.50	0	0	0	0	0	0	0	6.25	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	6.25	87.50	0	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	6.25	0	0	0	0	0	0	0	93.75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	6.25	93.75	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	6.25	0	93.75	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	6.25	0	0	0	0	0	0	0	0	0	93.75	0	0	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
ST	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.75	0	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
YY	0	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	93.75	0	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table A.32: Confusion matrix of the optimal classifier trained with dirfil4 features in Chinese dataset

	FS	***FS***	*FS*	FS	**HT**	***HT***	*HT*	HT	**KT**	***KT***	*KT*	KT	**LS**	***LS***	*LS*	LS	**ST**	***ST***	*ST*	ST	**YY**	***YY***	*YY*	YY
FS	93.75	0	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	87.50	0	0	0	0	0	0	6.25	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	6.25	0	93.75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	93.75	0	0	0	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	87.50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12.50	0	0
HT	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	6.25	0	0	0	0	0	0	0	93.75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	18.75	75.00	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
ST	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.75	0	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
YY	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	93.75	0	0	0
YY	0	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	93.75	0	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
YY	0	0	0	0	0	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	93.75

Table A.33: Confusion matrix of the optimal classifier trained with gabor features in Chinese dataset

	FS	***FS***	*FS*	FS	**HT**	***HT***	*HT*	HT	**KT**	***KT***	*KT*	KT	**LS**	***LS***	*LS*	LS	**ST**	***ST***	*ST*	ST	**YY**	***YY***	*YY*	YY
FS	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	93.75	0	0	0	0	0	0	0	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0
FS	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	0	93.75	0	0	0	0	0	0	0	6.25	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	6.25	0	0	0	0	0	93.75	0	0	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table A.34: Confusion matrix of the optimal classifier trained with LM features in Chinese dataset

	FS	***FS***	*FS*	FS	**HT**	***HT***	*HT*	HT	**KT**	***KT***	*KT*	KT	**LS**	***LS***	*LS*	LS	**ST**	***ST***	*ST*	ST	**YY**	***YY***	*YY*	YY
FS	93.75	0	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	87.50	0	6.25	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	6.25	0	0	0	0	0	0	0	93.75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	6.25	0	0	0	0	0	0	0	93.75	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	6.25	0	0	0	0	0	0	0	93.75	0	0	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	6.25	0	0	0	93.75	0	0	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.25	0	93.75	0	0	0	0	0
ST	0	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.75	0	0	0	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
YY	0	0	0	0	0	6.25	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	87.50	0	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table A.35: Confusion matrix of the optimal classifier trained with MR8 features in Chinese dataset

	FS	***FS***	*FS*	FS	**HT**	***HT***	*HT*	HT	**KT**	***KT***	*KT*	KT	**LS**	***LS***	*LS*	LS	**ST**	***ST***	*ST*	ST	**YY**	***YY***	*YY*	YY
FS	87.50	0	0	0	0	0	0	0	6.25	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0	0
FS	0	87.50	0	6.25	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	0	87.50	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.25	0
FS	0	0	6.25	93.75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	68.75	25.00	0	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	12.50	81.25	0	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0	0	0
HT	0	6.25	0	0	0	0	81.25	12.50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	12.50	0	0	0	0	0	0	0	50	31.25	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	50	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	0	0	81.25	18.75	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	0	0	12.50	87.50	0	0	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	68.75	31.25	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	6.25	6.25	0	0	0	0	0	0	37.50	50	0	0	0	0	0	0	0	0	0	0
LS	0	6.25	0	0	0	0	0	0	6.25	0	0	0	0	0	87.50	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	12.50	0	0	0	0	6.25	81.25	0	0	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	81.25	18.75	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	12.50	0	0	0	0	0	6.25	75.00	0	0	0	0	6.25	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87.50	6.25	0	0	6.25	0
ST	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.75	0	0	0	0
YY	0	0	0	0	0	0	6.25	0	0	0	0	0	6.25	0	0	0	0	0	0	0	81.25	0	6.25	0
YY	0	0	0	0	0	12.50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.25	81.25	0	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
YY	0	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.25	87.50

Table A.36: Confusion matrix of the optimal classifier trained with pyramid features in Chinese dataset

	FS	***FS***	*FS*	FS	**HT**	***HT***	*HT*	HT	**KT**	***KT***	*KT*	KT	**LS**	***LS***	*LS*	LS	**ST**	***ST***	*ST*	ST	**YY**	***YY***	*YY*	YY
FS	87.50	0	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0
FS	6.25	81.25	0	0	0	0	0	0	0	12.50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FS	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HT	0	6.25	0	0	0	0	0	93.75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	6.25	0	0	0	0	0	0	0	87.50	0	0	0	0	0	0	0	6.25	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	6.25	87.50	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
KT	0	0	0	6.25	0	0	0	0	0	0	0	93.75	0	0	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
LS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
ST	6.25	0	0	0	0	0	6.25	0	0	0	0	0	0	0	0	0	87.50	0	0	0	0	0	0	0
ST	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.75	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
ST	0	0	0	6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.75	0	0	0	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
YY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table A.37: Confusion matrix of the optimal classifier trained with CWT features in Farsi set

	H	***H***	*H*	H	**L**	***L***	*L*	L	**M**	***M***	*M*	M	**N**	***N***	*N*	N	**Ta**	***Ta***	*Ta*	Ta	**Tim**	***Tim***	*Tim*	Tim	*Titr*	Titr	**Tr**	***Tr***	*Tr*	Tr	**Y**	***Y***	*Y*	Y	**Z**	***Z***	*Z*	Z
H	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0
H	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0
L	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	19	0	0
L	0	0	0	0	0	0	0	69	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	25
M	0	0	0	0	0	0	0	0	81	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	94	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	6	0	0	0	88	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	6	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
Tim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tim	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Titr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
Titr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
Tr	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0
Tr	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	81	0	0	0	6	0	0	0	0	0	0
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	94	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	94	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
Z	0	0	0	0	25	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	63	0	0	0
Z	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88	0	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
Z	0	0	0	0	0	0	0	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	69

Table A.38: Confusion matrix of the optimal classifier trained with contourlet features in Farsi set

	H	***H***	*H*	H	**L**	***L***	*L*	L	**M**	***M***	*M*	M	**N**	***N***	*N*	N	**Ta**	***Ta***	*Ta*	Ta	**Tim**	***Tim***	*Tim*	Tim	*Titr*	Titr	**Tr**	***Tr***	*Tr*	Tr	**Y**	***Y***	*Y*	Y	**Z**	***Z***	*Z*	Z
H	81	0	13	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0
H	0	0	94	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0
L	0	0	0	0	44	6	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	19	0	6	0
L	0	0	0	0	19	44	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	6	6	0	6
L	0	0	0	0	0	0	38	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	19	0	
L	0	0	0	0	6	31	19	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
M	0	0	0	0	0	0	0	0	38	19	13	0	6	0	0	0	6	0	6	0	0	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	19	44	13	0	0	0	0	0	0	0	0	13	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0
M	0	0	0	0	0	0	6	0	0	6	50	19	6	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	13	63	0	6	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	81	6	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
N	0	0	0	0	0	0	0	0	0	13	0	0	13	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	6
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	56	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	44	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	88	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	63	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tim	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	13	69	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	6
Tim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	56	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tim	0	0	0	0	0	0	0	0	6	0	0	0	6	0	0	0	0	0	0	0	0	0	19	69	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Titr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88	6	6	0	0	0	0	0	0	0	0	0	0	0
Titr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	94	0	0	0	0	0	0	0	0	0	0	0	0
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	44	0	0	6	0	0	0	0	0	0	0
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	44	0	6	0	0	0	0	0	0	0	0
Tr	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	69	25	0	0	0	0	0	0	0	0
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	75	0	0	0	0	0	0	0	0
Y	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	6	0	0	44	25	0	0	6	0	0	0
Y	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	6	19	50	0	6	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	6	0	0	0	0
Y	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	88	0	0	0	0
Z	0	0	0	0	25	6	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	44	0	19	0
Z	0	0	0	0	6	6	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	63	0	6
Z	0	0	0	0	13	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	25	13
Z	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	6	69	

Table A.39: Confusion matrix of the optimal classifier trained with curvelet features in Farsi set

	H	***H***	*H*	H	**L**	***L***	*L*	L	**M**	***M***	*M*	M	**N**	***N***	*N*	N	**Ta**	***Ta***	*Ta*	Ta	**Tim**	***Tim***	*Tim*	Tim	*Titr*	Titr	**Tr**	***Tr***	*Tr*	Tr	**Y**	***Y***	*Y*	Y	**Z**	***Z***	*Z*	Z
H	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0
L	0	0	0	0	0	94	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	6	0	81	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0
L	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	88	0	6	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	6	0	0	0	81	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	6	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	6	0	0	0	0	0	0	0	0	0	88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0
Tim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tim	0	0	0	0	0	0	0	6	0	0	0	13	0	0	0	0	0	0	0	0	0	0	6	75	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Titr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	6	0	0	0	0	0	0	0	0	0	0	0
Titr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	88	0	0	0	0	0	0	0	0	0	0	0
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	6	0	0	0	0	0	0	0	0
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	88	6	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	94	0	0	0	0
Z	0	0	0	0	19	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	63	0	6	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	81	0	0
Z	0	0	0	0	13	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	69	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table A.40: Confusion matrix of the optimal classifier trained with dirfil3 features in Farsi set

	H	***H***	*H*	H	**L**	***L***	*L*	L	**M**	***M***	*M*	M	**N**	***N***	*N*	N	**Ta**	***Ta***	*Ta*	Ta	**Tim**	***Tim***	*Tim*	Tim	*Titr*	Titr	**Tr**	***Tr***	*Tr*	Tr	**Y**	***Y***	*Y*	Y	**Z**	***Z***	*Z*	Z
H	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0
L	0	0	0	0	88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0
L	0	0	0	0	0	88	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0
L	0	0	0	0	0	0	69	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	
L	0	0	0	0	0	0	0	88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	
M	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M	0	0	0	0	0	0	0	0	0	88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	
M	0	0	0	0	0	0	0	0	0	0	81	0	0	0	6	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	6	0	
M	0	0	0	0	0	0	0	0	0	0	0	88	0	0	0	6	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
N	0	0	0	0	6	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
N	0	0	0	0	0	6	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
N	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Tim	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Tim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Tim	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	81	13	0	0	0	0	0	0	0	0	0	0	0	0	0	
Tim	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	6	0	0	0	88	0	0	0	0	0	0	0	0	0	0	0	0	0	
Titr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	6	0	0	0	0	0	0	0	0	
Titr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88	0	6	0	6	0	0	0	0	0	0	
Tr	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	
Tr	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	
Y	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	
Z	0	0	6	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	81	0	0	
Z	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	
Z	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	6	0	75	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table A.41: Confusion matrix of the optimal classifier trained with dirfil4 features in Farsi set

	H	***H***	*H*	H	**L**	***L***	*L*	L	**M**	***M***	*M*	M	**N**	***N***	*N*	N	**Ta**	***Ta***	*Ta*	Ta	**Tim**	***Tim***	*Tim*	Tim	*Titr*	Titr	**Tr**	***Tr***	*Tr*	Tr	**Y**	***Y***	*Y*	Y	**Z**	***Z***	*Z*	Z
H	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0
H	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	6
L	0	0	0	0	6	81	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	6
L	0	0	0	0	0	0	75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0
L	0	0	0	0	0	6	6	63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	6	0	0	0	6	0
M	0	0	0	0	0	0	0	0	94	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	6	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	88	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	88	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0
Tim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tim	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tim	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	6	0	0	0	0	0	0	6	81	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Titr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
Titr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	6	0	0	0	0	0	0	0
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	81	6	0	0	6	0	0	0	6	0
Tr	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	6	0	0	0
Y	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0
Y	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0
Y	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0
Z	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	81	0	0	0	0
Z	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0
Z	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	6	0	56	0	0
Z	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	88

Table A.42: Confusion matrix of the optimal classifier trained with gabor features in Farsi set

	H	***H***	*H*	H	**L**	***L***	*L*	L	**M**	***M***	*M*	M	**N**	***N***	*N*	N	**Ta**	***Ta***	*Ta*	Ta	**Tim**	***Tim***	*Tim*	Tim	*Titr*	Titr	**Tr**	***Tr***	*Tr*	Tr	**Y**	***Y***	*Y*	Y	**Z**	***Z***	*Z*	Z
H	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0
L	0	0	0	0	0	88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	6	0	0	0
L	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0
L	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0
M	0	0	0	0	0	0	0	0	94	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	81	0	0	0	13	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	6	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0
Tim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tim	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Titr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
Titr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	6	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
Z	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	88	0	0	0
Z	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
Z	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	81

Table A.43: Confusion matrix of the optimal classifier trained with LM features in Farsi set

	H	***H***	*H*	H	**L**	***L***	*L*	L	**M**	***M***	*M*	M	**N**	***N***	*N*	N	**Ta**	***Ta***	*Ta*	Ta	**Tim**	***Tim***	*Tim*	Tim	*Titr*	Titr	**Tr**	***Tr***	*Tr*	Tr	**Y**	***Y***	*Y*	Y	**Z**	***Z***	*Z*	Z
H	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0
H	0	0	94	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0
L	0	0	0	0	50	6	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	19	0	
L	0	0	0	0	0	50	0	13	0	0	0	0	0	6	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	6	0	0	0	6	6	6	
L	0	0	0	0	13	0	56	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	19	0		
L	0	0	0	0	0	19	13	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	13		
M	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
M	0	0	0	6	0	0	0	0	6	88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
M	0	0	0	0	0	0	0	0	0	0	94	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
M	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	6		
N	0	0	0	0	0	0	0	0	6	0	0	0	75	0	0	0	0	0	6	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	6	0		
N	0	0	0	0	0	6	0	0	0	13	0	0	0	75	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0		
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Tim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Tim	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	13	75	0	0	0	0	0	0	0	0	0	0	0	0	6	0		
Tim	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	88	0	0	0	0	0	0	0	0	0	0	0	0	0		
Tim	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	6	69	0	0	0	0	0	0	0	0	0	0	0	0		
Titr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0		
Titr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0		
Tr	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	69	0	0	0	13	0	0	0	0	0		
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0		
Tr	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	69	6	0	0	6	0	0	0	6	0
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	6	0	0	0	0
Y	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	75	0	0	0	6	6	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	81	0	6	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	75	13	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	6	0	88	0	0	0
Z	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	6	0	0	6	0	0	0	50	6	6	0
Z	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	81	0	6
Z	0	0	0	0	0	0	38	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	6	19	0	13	13
Z	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	88

Table A.44: Confusion matrix of the optimal classifier trained with MR8 features in Farsi set

	H	***H***	*H*	H	**L**	***L***	*L*	L	**M**	***M***	*M*	M	**N**	***N***	*N*	N	**Ta**	***Ta***	*Ta*	Ta	**Tim**	***Tim***	*Tim*	Tim	*Titr*	Titr	**Tr**	***Tr***	*Tr*	Tr	**Y**	***Y***	*Y*	Y	**Z**	***Z***	*Z*	Z
H	13	63	6	6	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0
H	44	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	13	0	6	0	0	0	0	0	0	0	0	0	0
H	0	0	44	13	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	6	0	0	6	0	0	0	0
H	6	0	6	56	6	0	0	0	0	0	0	0	0	6	6	0	0	0	0	0	0	0	0	0	6	0	0	0	0	6	0	0	0	0	0	0	0	0
L	0	0	0	0	25	56	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	6	0	0
L	0	0	0	0	50	19	6	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	6	6	0
L	0	0	0	0	0	0	38	13	0	0	6	0	6	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	13	13	0
L	0	0	0	0	6	13	6	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	6	0	0	0	25	0
M	0	0	0	0	0	0	0	0	56	31	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0
M	0	0	0	13	0	0	0	0	19	31	0	0	13	13	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	6	0	38	19	6	0	0	0	0	0	6	13	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	6	0	0
M	0	0	0	0	0	0	0	0	0	6	25	38	6	0	13	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0
N	0	0	0	6	6	0	0	0	0	13	6	0	25	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	6	6	0	0
N	6	0	6	6	0	0	0	0	0	6	0	13	13	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0
N	0	0	0	0	0	0	13	6	0	0	0	6	0	0	38	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31	56	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	44	50	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	38	38	0	0	0	0	0	0	6	0	0	0	0	13	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	6	6	6	0	0	0	0	0	75	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tim	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	44	38	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	6	6	0	56	13	0	0	0	0	0	0	0	0	6	0	0	0	0	0	6	0
Tim	0	0	0	0	0	0	0	0	0	0	6	6	0	0	6	0	0	0	0	0	13	0	38	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tim	0	0	0	0	0	0	0	0	0	0	6	6	0	0	6	0	0	0	0	0	0	31	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Titr	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	69	25	0	0	0	0	0	0	0	0	0	0	0	0
Titr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	75	0	0	0	0	0	0	0	0	0	0	0	0
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	75	0	0	13	0	0	0	0	6	0	0
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	81	19	0	0	0	0	0	0	0	0	0	0
Tr	0	0	6	0	0	0	0	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	19	25	6	0	13	6	6	0	0	0
Tr	0	0	0	0	0	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	50	0	6	0	6	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	19	0	19	0	19	13	6	0	13	6	0	0
Y	0	0	0	0	0	13	0	6	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	19	0	6	6	13	31	0	0	0	0	0	0
Y	0	0	0	0	13	0	6	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	31	38	0	0	0	0
Y	0	0	0	0	0	0	6	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	63	6	0	0	0	0
Z	0	0	6	0	13	0	0	0	6	6	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	6	0	0	13	0	13	19	6	6
Z	0	0	6	0	0	6	6	6	0	0	0	0	6	13	0	0	0	0	0	0	0	0	0	0	6	0	0	6	0	0	19	0	0	0	6	19	0	0
Z	0	0	0	0	0	0	19	19	6	0	6	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	25	0	0	13
Z	0	0	0	0	0	0	13	19	0	0	0	6	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	13	25

Table A.45: Confusion matrix of the optimal classifier trained with pyramid features in Farsi set

	H	***H***	*H*	H	**L**	***L***	*L*	L	**M**	***M***	*M*	M	**N**	***N***	*N*	N	**Ta**	***Ta***	*Ta*	Ta	**Tim**	***Tim***	*Tim*	Tim	*Titr*	Titr	**Tr**	***Tr***	*Tr*	Tr	**Y**	***Y***	*Y*	Y	**Z**	***Z***	*Z*	Z
H	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	56	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	25	0	0	0	0
L	0	0	0	0	0	69	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	13	
L	0	0	0	0	19	0	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	6	
L	0	0	0	0	0	13	6	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	38	
M	0	0	0	0	0	0	0	0	81	0	0	6	0	0	0	0	0	6	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	
M	0	0	0	0	0	0	0	0	0	63	0	6	0	13	0	0	0	6	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	6	
M	0	0	0	0	0	0	0	0	0	0	75	0	6	0	6	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M	0	0	0	0	0	0	0	0	0	0	0	88	0	0	0	6	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	
N	0	0	6	0	0	0	0	0	6	0	0	0	81	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	
N	0	0	0	6	0	0	0	0	0	19	0	0	0	63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	
N	0	0	0	0	0	0	0	0	0	0	6	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
N	0	0	0	0	0	0	0	0	0	0	0	6	0	0	6	81	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	
Ta	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Tim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Tim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	75	0	0	0	0	0	0	0	0	0	0	0	6	6	0	0	
Tim	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Tim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	
Titr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	6	0	0	0	0	0	0	
Titr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	6	0	0	0	0	0	0	0	0	0	
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	81	0	6	0	13	0	0	0	0	0	0	
Tr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	6	0	0	0	0	0	
Tr	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	19	0	0	0	0	0	0	0	
Tr	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	0	0	0	
Y	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	6	0	0	6	0	0	0	0	0	56	0	6	0	13	0	0	
Y	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	81	0	6	0	0	0	
Y	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	6	0	0	6	6
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
Z	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	6	0	0	0	0	69	0	6	0
Z	0	0	0	6	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0
Z	0	0	0	0	0	0	25	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	13	0	44	6
Z	0	0	0	0	0	0	0	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	63	

Table A.46: Confusion matrix of the optimal classifier trained with CWT features in ALPH-REGIM dataset

	Ahsa	Andalus	Arabictransparent	Badr	Burdiyah	Dammam	Hada	Kharj	Koufi	Naskh
Ahsa	99.69	0.06	0.06	0	0.12	0	0.06	0	0	0
Andalus	0.15	98.77	0	0.08	0	0	0.15	0.77	0	0.08
Arabictransparent	0	0	99.65	0	0.27	0	0	0	0	0.09
Badr	0	0	0	99.68	0	0.08	0.16	0.08	0	0
Burdiyah	0.11	0	0	0	97.84	0.11	0	0	0	1.93
Dammam	0.09	0.09	0	0.09	0	99.72	0	0	0	0
Hada	0.05	0.05	0	0.05	0.11	9.28	90.22	0.11	0.05	0.05
Kharj	0.12	0.41	0	0	0.12	9.42	0	89.88	0	0.06
Koufi	0	0.11	0	0.11	0.22	0.22	0	0	99.24	0.11
Naskh	0	0	0	0	1.65	0	0.11	0.11	0	98.13

Table A.47: Confusion matrix of the optimal classifier trained with contourlet features in ALPH-REGIM dataset

	Ahsa	Andalus	Arabictransparent	Badr	Burdiyah	Dammam	Hada	Kharj	Koufi	Naskh
Ahsa	98.72	0.37	0.06	0.06	0	0.31	0.24	0.12	0	0.12
Andalus	0.61	95.63	0.08	0.38	0.23	0.08	0.15	1.99	0.31	0.54
Arabictransparent	0	0	99.47	0	0.18	0.35	0	0	0	0
Badr	0.48	1.69	0	96.69	0.08	0.40	0.24	0.08	0.08	0.24
Burdiyah	0.11	0.23	0.34	0.34	94.77	0.91	0	0.11	0.57	2.61
Dammam	0.05	0.23	0	0.14	0.05	99.06	0.09	0.14	0.09	0.14
Hada	0.27	0.05	0	0.27	0.11	9.39	88.75	0.98	0.05	0.11
Kharj	0	1.76	0	0	0.23	9.54	0.35	88.12	0	0
Koufi	0	0.33	0.22	0.22	0.76	0.76	0	0	97.39	0.33
Naskh	0.33	0.77	0.22	0.22	3.85	1.21	0	0.33	0.33	92.75

Table A.48: Confusion matrix of the optimal classifier trained with curvelet features in ALPH-REGIM dataset

	Ahsa	Andalus	Arabictransparent	Badr	Burdiyah	Dammam	Hada	Kharj	Koufi	Naskh
Ahsa	99.45	0.12	0.06	0	0.12	0.06	0.18	0	0	0
Andalus	0.08	98.24	0	0.08	0.08	0	0.23	1.30	0	0
Arabictransparent	0	0	99.82	0	0	0	0	0	0	0.18
Badr	0	0.16	0	99.35	0.08	0	0.24	0.08	0	0.08
Burdiyah	0	0	0.23	0	97.61	0.11	0	0.11	0.34	1.59
Dammam	0	0.14	0	0	0	99.81	0	0.05	0	0
Hada	0	0.27	0	0.05	0	9.28	90.01	0.11	0	0.27
Kharj	0.18	0.88	0	0	0.06	9.60	0.23	89.06	0	0
Koufi	0	0	0.11	0	0.87	0.22	0.22	0	98.48	0.11
Naskh	0.22	0.11	0	0	1.54	0.22	0.22	0.11	0.11	97.47

Table A.49: Confusion matrix of the optimal classifier trained with dirfil3 features in ALPH-REGIM dataset

	Ahsa	Andalus	Arabictransparent	Badr	Burdiyah	Dammam	Hada	Kharj	Koufi	Naskh
Ahsa	99.20	0.31	0	0.06	0.12	0	0.06	0	0.06	0.18
Andalus	0.54	97.70	0	0	0.23	0	0.61	0.92	0	0
Arabictransparent	0	0	99.47	0	0.27	0	0	0.18	0.09	0
Badr	0.08	0.24	0	99.19	0	0	0.08	0.08	0.24	0.08
Burdiyah	0	0.23	0.34	0	97.84	0.11	0.23	0	0.23	1.02
Dammam	0.09	0.09	0	0.19	0.05	99.48	0	0	0.09	0
Hada	0.27	0.33	0	0.16	0.38	9.34	88.69	0.05	0	0.76
Kharj	0.06	0.94	0.18	0	0.06	9.42	0.18	89.06	0	0.12
Koufi	0	0.11	0.11	0.54	0.11	0.33	0.33	0	98.37	0.11
Naskh	0	0.33	0	0.11	1.32	0.11	1.32	0.44	0	96.37

Table A.50: Confusion matrix of the optimal classifier trained with dirfil4 features in ALPH-REGIM dataset

	Ahsa	Andalus	Arabictransparent	Badr	Burdiyah	Dammam	Hada	Kharj	Koufi	Naskh
Ahsa	98.90	0.24	0.06	0.12	0.12	0	0.37	0.06	0	0.12
Andalus	1.23	97.24	0	0.23	0	0.08	0.08	1.15	0	0
Arabictransparent	0.09	0	99.65	0	0.18	0	0	0	0	0.09
Badr	0.32	0.32	0	98.95	0	0	0.16	0	0	0.24
Burdiyah	0	0	0.11	0	98.52	0.11	0.23	0.11	0.11	0.80
Dammam	0	0.28	0	0	0	99.72	0	0	0	0
Hada	0.60	0.11	0	0.16	0.27	9.28	89.40	0.11	0	0.05
Kharj	0.18	0.88	0	0	0.06	9.48	0.06	89.29	0	0.06
Koufi	0	0	0.11	0.22	0	0.22	0	0	99.24	0.22
Naskh	0.33	0.11	0.22	0.44	0.44	0.22	0.77	0.11	0	97.36

Table A.51: Confusion matrix of the optimal classifier trained with gabor features in ALPH-REGIM dataset

	Ahsa	Andalus	Arabictransparent	Badr	Burdiyah	Dammam	Hada	Kharj	Koufi	Naskh
Ahsa	99.69	0	0	0	0.06	0	0.06	0.12	0	0.06
Andalus	0	99.31	0.08	0	0	0	0.15	0.31	0	0.15
Arabictransparent	0.09	0	99.56	0	0	0	0.18	0	0	0.18
Badr	0	0	0	99.84	0	0	0	0.08	0	0.08
Burdiyah	0	0	0	0	97.95	0.11	0.68	0	0.23	1.02
Dammam	0	0	0	0.05	0	99.95	0	0	0	0
Hada	0.05	0	0.05	0.05	0.16	9.28	90.17	0.05	0.05	0.11
Kharj	0.06	0.12	0	0.06	0	9.36	0.18	89.99	0	0.23
Koufi	0	0	0.11	0.11	0.22	0.22	0	0	99.35	0
Naskh	0.11	0	0.11	0.11	0.66	0	0	0.11	0	98.90

Table A.52: Confusion matrix of the optimal classifier trained with LM features in ALPH-REGIM dataset

	Ahsa	Andalus	Arabictransparent	Badr	Burdiyah	Dammam	Hada	Kharj	Koufi	Naskh
Ahsa	98.96	0.31	0.06	0	0.18	0.18	0.06	0.06	0.12	0.06
Andalus	0.38	96.32	0.08	0	0.15	0.08	0.23	2.61	0.08	0.08
Arabictransparent	0.18	0	99.38	0	0.27	0	0	0.09	0	0.09
Badr	0.40	0.40	0	98.31	0.16	0.16	0.08	0.08	0	0.40
Burdiyah	0.23	0.11	0.11	0	98.41	0.11	0	0.11	0.45	0.45
Dammam	0.33	0.05	0.19	0.05	0	99.39	0	0	0	0
Hada	0.38	0.16	0	0.22	0.16	9.28	89.51	0.05	0	0.22
Kharj	0.41	1.29	0.06	0	0.06	9.48	0.12	88.53	0	0.06
Koufi	0	0.11	0.11	0.11	0.33	0.87	0.22	0	97.83	0.43
Naskh	0.33	0.22	0.44	0.44	0.66	0.44	0.22	0.11	0.33	96.81

Table A.53: Confusion matrix of the optimal classifier trained with MR8 features in ALPH-REGIM dataset

	Ahsa	Andalus	Arabictransparent	Badr	Burdiyah	Dammam	Hada	Kharj	Koufi	Naskh
Ahsa	0	0	0	0	0	0	0	39.02	58.17	2.81
Andalus	0	0	0	0	0	0	0	74.85	23.62	1.53
Arabictransparent	0	0	0	0	0	0	0	99.73	0.18	0.09
Badr	0	0	0	0	0	0	0	51.77	47.10	1.13
Burdiyah	0	0	0	0	0	0	0	1.82	94.43	3.75
Dammam	0	0	0	0	0	0	0	79.77	10.14	10.09
Hada	0	0	0	0	0	0	0	6.61	83.67	9.72
Kharj	0	0	0	0	0	0	0	87.13	3.10	9.77
Koufi	0	0	0	0	0	0	0	0.65	98.37	0.98
Naskh	0	0	0	0	0	0	0	1.76	7.91	90.33

Table A.54: Confusion matrix of the optimal classifier trained with pyramid features in ALPH-REGIM dataset

	Ahsa	Andalus	Arabictransparent	Badr	Burdiyah	Dammam	Hada	Kharj	Koufi	Naskh
Ahsa	99.27	0.18	0.06	0.06	0.06	0	0.24	0.12	0	0
Andalus	0.23	97.32	0	0	0	0.15	0.23	2.07	0	0
Arabictransparent	0	0	99.56	0	0.27	0.09	0	0.09	0	0
Badr	0.32	0.48	0.08	98.55	0	0.24	0.24	0	0	0.08
Burdiyah	0	0	0	0	98.86	0.11	0.34	0.11	0.34	0.23
Dammam	0	0.09	0.09	0.09	0.05	99.62	0	0.05	0	0
Hada	0.55	0.16	0	0.11	0.16	9.28	89.46	0.05	0.05	0.16
Kharj	0.18	1.58	0	0	0	9.48	0	88.77	0	0
Koufi	0	0	0	0.11	0.43	0.54	0.22	0	98.59	0.11
Naskh	0	0.22	0	0.11	0.77	0.55	0.33	0.99	0.22	96.81